

Received December 18, 2019, accepted January 10, 2020, date of publication January 14, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2966497

SlimRGBD: A Geographic Information Photography Noise Reduction System for Aerial Remote Sensing

CHUNXUE WU^{ID}¹, (Member, IEEE), BOBO JU^{ID}¹, YAN WU^{ID}²,
AND NAI XUE XIONG^{1,3}, (Senior Member, IEEE)

¹School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

²O'Neill School of Public and Environmental Affairs, Indiana University Bloomington, Bloomington, IN 47405, USA

³Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, CO 74464, USA

Corresponding author: Chunxue Wu (wxc@usst.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0810204, in part by the National Natural Science Foundation of China under Grant 61872242 and Grant 61502220, in part by the Shanghai Science and Technology Innovation Action Plan Project under Grant 17511107203 and Grant 16111107502, and in part by the Shanghai Key Lab of Modern Optical System.

ABSTRACT In the past ten years, civil drone technology has developed rapidly, and UAV (Unmanned Aerial Vehicle) has been widely used in various industries. Especially in the field of aerial remote sensing, the emergence of UAV technology has enabled the geographical information of remote areas that are not concerned to be quickly presented. However, UAV aerial photography is greatly affected by the weather. Pictures that use aerial drones for aerial photography in rainy weather will appear noise. In this paper, how to eliminate the noise of aerial image is to be talked, the multi-channel pruning technology is used to pruning the RnResNet network. Based on this, a new anti-convergence-convolution neural network noise reduction system for the operation of UAV airborne embedded equipment is proposed. The system is used to eliminate noise in the aerial image. This type of noise reducer has got rid of the current situation that the neural network noise reducer consumes too much power and is inefficient, and has certain advantages.

INDEX TERMS SlimRGBD, ResNet, generative adversarial networks, image noise reduction, UAV, channel pruning, sparse training.

I. INTRODUCTION

UAV technology has been an important part of human early aviation and has a history of more than 100 years. At present, many countries have adopted drone technology as a frontier technology. Civil drone applications have less than one decade. In modern society, remote sensing technology has become an important means for humans to acquire geographical environment and change information [1]. With the advent of the information age, the demand for remote sensing data has increased dramatically in various countries. Existing satellite remote sensing and aerial remote sensing technologies have the characteristics of obtaining large-scale macro geographic information [2]. However, for many remote sensing technology applications with high resolution requirements and fast update time, it is difficult to guarantee. The use

of drones as remote sensing platforms for aerial photography and ground observations provides a new technical approach to this emergency response [3]. With the development and improvement of low-altitude drone photogrammetry technology, a large number of experiments have shown that the accuracy of UAV mapping topographic map can meet the requirements of 1:2000 topographic map.

UAV as an aerial remote sensing platform has the following advantages: (1) Fast maneuvering response capability. The low-altitude drone system has a short lift-off time, simple operation, and convenient transportation, and can quickly reach the monitoring area. Fly fast and photography quickly [4]. (2) High resolution images and high precision positioning capabilities. The spatial resolution of the image acquired by the system reached the decimeter level. They are high-resolution digital images acquired for high-resolution 3D landscape maps [1]. (3) The cost of using a drone is low. UAVs are inexpensive to design and produce because

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy^{ID}.

they do not install flight crew driving equipment, voice communications, and security equipment. The highly integrated design of the drones due to the universal application of digital technology makes the production cost low. UAVs are inexpensive to produce tooling and materials due to the ability to properly reduce safety requirements and allow for the extensive use of composite materials and their new manufacturing processes [5]. (4) UAV can undertake high-risk or high-tech task flights. Drivers or researchers can work safely on the ground. Flight does not cause accidents due to human error or flight measurement failure. When conducting real-time information research, the number of people working is not limited. Long-term or continuous real-time data transmission, real-time and dynamic of fidelity. Especially for environmental monitoring that cannot be reached by vehicles and ships, environmental monitoring in toxic areas, disaster monitoring and command and rescue, the UAV remote sensing system can show its unique advantages [6].

However, most of the aerial surveys of drones are subject to weather and light conditions. In the environment with good weather and sufficient sunlight, the quality of remote sensing images is higher. Remote sensing images with poor shooting conditions on rainy or weak lighting conditions are of poor quality. This is because light good or bad is the main influencing factor for shooting quality of the images [7]. Under low light conditions, the image taken by the drone camera is likely to form a noisy image. It mainly refers to the rough part of the image generated by CCD (CMOS) in the process of receiving and outputting light as the receiving signal. Digital photos taken by drones may not be noticed if they are reduced by a personal computer, if you magnify the image, then there will be a color (false color) that is not present. This false color is image noise. The cause of the noise generated by the drone camera: (1) Image noise caused by long exposure. This phenomenon mainly occurs in low-light shooting, and in the dark areas of the image, some isolated bright spots appear. It can be said that the reason is that the CCD cannot handle the huge workload caused by the slow shutter speed, causing some specific pixels to lose control. (2) Image noise generated by compressing an image in JPEG format. Since images in JPEG format still appear natural after reducing the image size, special methods can be used to reduce image data. At this point, it will process pixels in a row above and below. Therefore, especially at the edge of the pixel edge, an unnatural combination with the next pixel unit occurs. Image noise generated by compression in JPEG format is also called Block Noise. The higher the compression ratio, the more obvious the image noise. Although the noise becomes invisible after the image is reduced, the color compensation is very obvious. Such image noise can be solved by using the highest possible image quality or by recording images other than the JPEG format. (3) Image noise caused by blur filtering. Image noise caused by fuzzy filtering, like JPEG, causes image noise when processing images. Sometimes it is generated during the internal processing of a digital camera, sometimes when

it is processed by image retouching software. For smaller images, image noise is produced in order to make the image appear sharper and emphasize its color edges [8].

GAN have proven to be outstanding in image noise reduction. However, compared with UAV equipment, the common GAN method for image noise reduction requires high hardware resources of the device, but the memory and computing power of UAV embedded devices are limited. It is a big challenge to use the GAN method to achieve image denoising on embedded devices with limited computing power while ensuring superior noise reduction performance. The goal of this study was to develop a model that would allow drones to produce realistic images even after processing aerial images with noisy images. The main challenges are twofold: First, the model of this study should be flexible and robust to handle the same image corrupted by different levels of noise; second, we must ensure that the denoised image is real and visually pleasing. To solve these challenges, we propose a new type of computing lightweight SlimRGBD (Slim ReCNN-GAN Blind Denoising) for image noise reduction on UAV.

The rest of this paper is organized as follows: In the second section, this article will introduce the efforts and shortcomings of the previous research on image noise reduction, and the problems that this article will try to solve. In the third chapter, the paper elaborates the model design and system principle of the image noise reducer in this study. The fourth chapter describes in detail the experimental process of this study, as well as the comparison of the methods used in this study with the historical advanced methods. The fifth chapter summarizes some of the characteristics of this study and the work to be carried out in the future.

II. RELATED WORK

A. LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORK FOR UAV VISUAL RECOGNITION

UAVs or Universal UAVs, which are computer vision enabled by onboard cameras and embedded systems, are popular in a wide range of applications. However, due to the limited memory and computing power of embedded devices, real-time scene analysis by target detection running on a drone platform is very challenging. In order to solve the problem of reducing the memory footprint of the ResNet type convolutional network architecture, Pierre Stock et al. proposed a vector quantization method. The goal is to maintain the quality of the network output refactoring, not the weight. The advantage of the vector quantization method is that it minimizes the loss reconstruction error of the input in the domain and does not require any tag data. Pierre Stock et al. also used a byte-aligned codebook to generate a compressed network for efficient reasoning on the CPU [9]. The vector quantization method was verified by quantifying a high-performance ResNet-50 model to a memory size of 5 mb (20 compression factors) while maintaining a 76.1% top-1 accuracy on the ImageNet object classification. The method can be easily adapted to simultaneously compress and transfer ResNet trained on ImageNet to other domains.

However, no consideration is given to nonlinearity, and there is a certain range of reconstruction errors.

As we all know, power consumption is always positively related to the size of the function implemented by the application. UAV applications usually require low power consumption to ensure the endurance of the drone. In order to solve these problems, Pengyi Zhang et al. proposed a highly efficient target detector on the UAV through channel pruning of the convolutional layer [10]. They enhanced the channel-level sparsity of the convolutional layer by applying L1 regularization to the channel scale factor, and pruning the feature channel with less information to obtain a sparse target detector. Based on this method, Pengyi Zhang et al. proposed SlimYOLOv3, which has fewer promising parameters and floating point operations (FLOPs) than the original YOLOv3. It is a promising real-time target detection solution for drones. They evaluated SlimYOLOv3 on the VisDrone2018-Det benchmark dataset. SlimYOLOv3 achieved convincing results compared to unpruned convolutional neural networks, with FLOPs reduced by approximately 90.8% and parameter sizes reduced by approximately 92.0%. The running speed is twice that of YOLOv3 [11], and the detection accuracy is also quite good. This proves that the L1 regularization is applied to the channel scale factor to enhance the channel-level sparsity of the convolutional layer, and the convolutional neural network obtained by pruning the feature channel with less information is more efficient, faster and better. It is more suitable for the realization of tasks such as image denoising using the deep learning method on the drone.

B. ADVANTAGES AND DISADVANTAGES OF MAINSTREAM METHODS FOR IMAGE NOISE REDUCTION

Existing image denoising methods mostly assume Gaussian white noise of known intensity with a uniform Gaussian distribution of noise. However, in real noise images, the noise model is usually unknown and may be more complicated. In response to this problem, Fengyuan Zhu et al. proposed a new blind image denoising algorithm for recovering clean images from noise images containing unknown noise models [12]. In order to model the empirical noise of an image, their method introduces a mixture of Gaussian distributions that is flexible enough to approximate different continuous distributions. Redefine the blind image denoising problem as a learning problem. First, a two-layer structural model of noise plaque is established, and the net spot is used as a latent variable. In order to control the complexity of the noise plaque model, Fengyuan Zhu et al. proposed a new Bayesian nonparametric prior, that is, relying on the Dirichlet process tree to build the model [12]. Then, a variational inference algorithm is derived to estimate the model parameters and restore a clean patch. This method is applied to the synthesis and real noise images of different noise models.

Convolutional neural networks have always been a research hotspot to solve image denoising problems, but their performance is still not satisfactory in most applications.

The synthetic noise distribution of these network trainings does not accurately reflect the noise captured by the image sensor. Data sets for some clean noise image pairs are often used for benchmarking or for specific applications. The main reason is that the model being studied is easily over-fitting on the simplified AWGN model, which seriously deviates from the complex actual noise model. In order to improve the generalization ability of deep CNN denoiser, Shi Guo et al. proposed a method of training convolutional blind denoising network (CBDNet) [13]. In order to further provide an interactive strategy to easily correct the denoising results, they embedded a noise estimation sub-network with asymmetric learning into the CBDNet to suppress low estimation of noise levels. The main findings of this work are twofold. First, real noise models, including miscellaneous Gaussian and ISP pipelines, are key to making composite image learning models suitable for real-noise photos. Secondly, adding synthetic and real noise images to the training can improve the denoising performance of the network. Benoit Brummer et al. introduced the Natural Image Noise Data Set (NIND), a dataset of SLR-like images with different ISO noise levels, which is large enough to train the model for blind noise reduction over a wide range of noise [14]. They demonstrated a denoising model trained with NIND and showed that it is significantly better than BM3D in terms of ISO noise from unseen images, even when it is extended to images from different types of cameras. Most of Benoit Brummer's experiments are conducted on a single U-Net [15], [16] network, but modern methods, such as conditionally generated confrontational networks (GANs), may yield better performance.

Kai Zhang et al. took a step more forward by studying the construction of feedforward denoising of convolutional neural networks (DnCNNs), which introduced a very deep architecture, learning algorithms and regularization methods into image denoising [17]. Specifically, the residual learning and batch normalization are applied to accelerate the denoising model training process and promote the performance of image denoising. Their proposed DnCNN model is capable of dealing with Gaussian noise (ie, Gaussian white noise) denoising at unknown noise levels, blind Gaussian denoising). Using the residual learning strategy, DnCNN implicitly removes hidden images hidden in the hidden layer. This feature motivates them to train a DnCNN model to deal with some common image denoising tasks such as Hybrid Noise denoising, single image super resolution, and JPEG format image denoising. This method not only has good image denoising performance in quantitative and qualitative, but also has good running timeliness in a device with a GPU implementation [17].

Based on discriminative learning DnCNN, the most advanced denoising effect can be achieved, but these methods are not suitable for this problem due to the lack of paired training data. To tackle this problem, Jingwen Chen et al. proposed a two-step solution: (1) The training generates a Generative Adversarial Network (GAN) to estimate the noise distribution on the input noise image to generate the

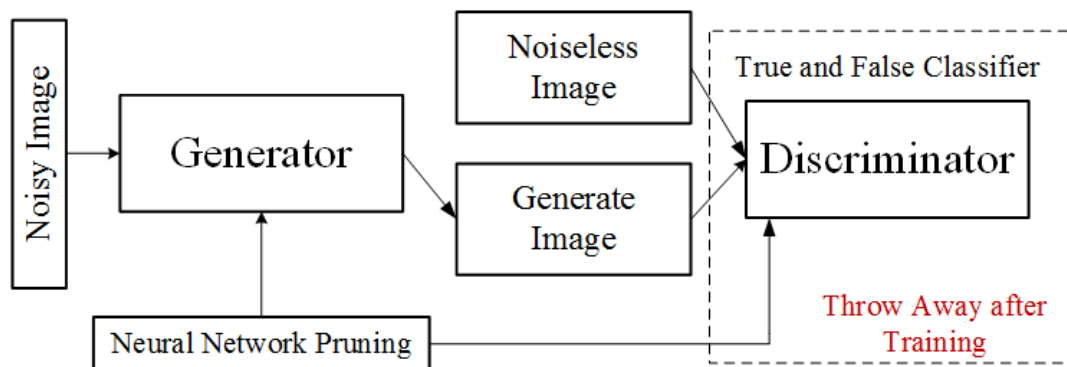


FIGURE 1. SlimRGBD method framework. “Generator” is a neural network for generating noise-reduced images, and “Discriminator” is a convolutional neural network for determining the quality of noise-reduced images. “Generator” and “Discriminator” have different neural network architectures.

noise samples. (2) The noise block generated by the first step is constructed to denoise the training datasets and the training deep convolutional neural network (CNN) [18]. They attempted to improve the performance of image blind noise reduction using a deep learning-based approach in the absence of paired training data. On this basis, they proposed the GCBD algorithm [18]. The GAN is used to learn the noise distribution, establish a paired training dataset, and train the CNN for denoising. One limitation of this approach is to assume that the noise is zero mean additive noise. This type of noise is common in the natural environment and includes a wide variety of noise.

Majed El Helou et al. proposed a theory-based Gaussian noise blind learning general image denoiser (blind universal image fusion denoising network, BUIFD) [8]. This method is called fusion denoising. It has a strong generalization ability for invisible noise levels. Their approach improves the PSNR of real-world grayscale image denoising with up to 0.7 dB of training noise levels. It also improves the single-noise performance of each of the most advanced color image denoising performance, averaging 0.1 decibels.

III. SYSTEM MODEL AND DEFINITIONS

The new SlimRGBD (Slim ReCNN-GAN Blind Denoising) we proposed for UAV image noise reduction can be seen semantically that our SlimRGBD model uses: ReCNN, neural network punning for ReCNN, GAN, blind noise reduction and other technologies. A simple framework diagram of the entire method is as Figure 1. This section will explain in detail how these technologies are applied in SlimRGBD.

A. GENERATIVE ADVERSARIAL NETWORK

The main inspiration of GAN comes from the idea of zero-sum game in game theory [19]. When applied to deep neural networks, it is to generate G (Generator) and D (Discriminator) to continuously play, so that G learns the distribution of data. If image generation is used, G can generate realistic images from a random number after training

is completed [20]. The main functions of G and D are: (1) G is a generative network that receives a random noise z (random number) and generates an image through this noise; (2) D is a discriminant network that discriminates whether the picture is “real”. Its input is x , which represents a picture, and the output $D(x)$ represents the probability of a real picture. If $D(x)$ is 1, it means that it is a real picture, and if $D(x)$ is 0, it means that it is impossible to be a real picture [21].

Compared with the traditional model, it has two different networks instead of a single network, and the training method adopts the gradient update information of G in the confrontation training mode GAN from the discriminator D, instead of from the data sample [22]. GAN is a generative model that uses only backpropagation compared to other generation models (Boltzmann machines and GSNs [23]) without the need for complex Markov chains. Compared to all other models, GAN can produce clearer, more realistic samples [24]. GAN uses an unsupervised learning style training that can be widely used in unsupervised learning and semi-supervised learning. Compared to variational self-encoders, GANs does not introduce any deterministic bias, variational methods. Deterministic biases are introduced because they optimize the lower bound of the log likelihood rather than the likelihood itself, which seems to result in instances of VAEs being generated more ambiguous than GANs. Compared with VAE, GANs has no variation lower bound [25]. If the discriminator is well trained, the generator can perfectly learn the distribution of training samples. In other words, GANs are gradual, but VAE is biased. GAN is applied to some scenes, such as image style migration, super resolution, incomplete image, denoising, avoiding the difficulty of loss function design, no matter what, as long as there is a benchmark, you can use the discriminator directly, and then leave the rest The task is handed over to the confrontation training. Training GAN needs to achieve Nash equilibrium, sometimes it can be done with gradient descent, sometimes it can’t. There is no best Nash equilibrium method, so training GAN is unstable compared to VAE or PixelRNN, but in practice it is much more stable than training Boltzmann [25].

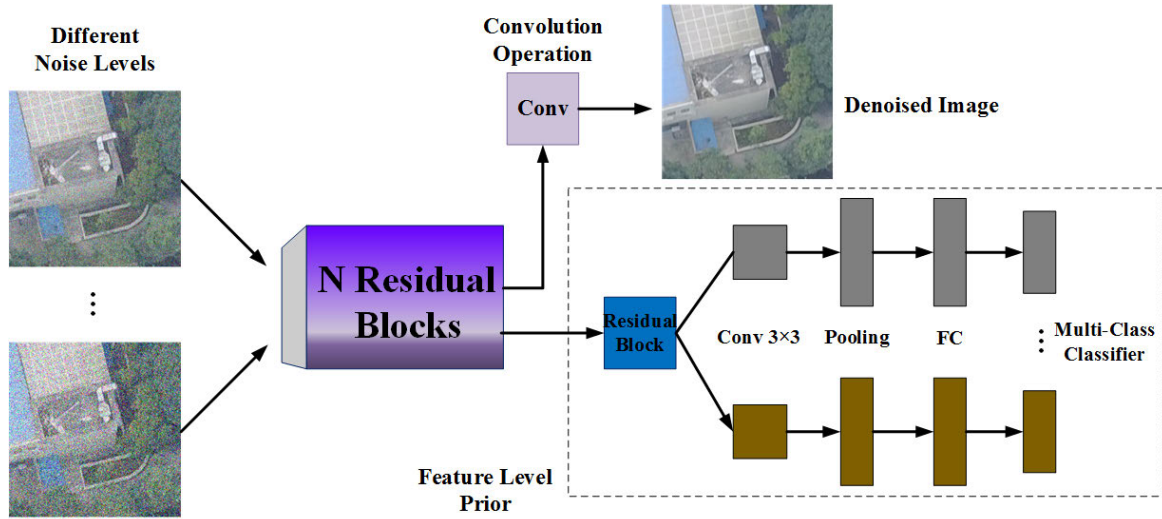


FIGURE 2. The dotted rectangle is applied to build the loss function to learn the feature-class prior.

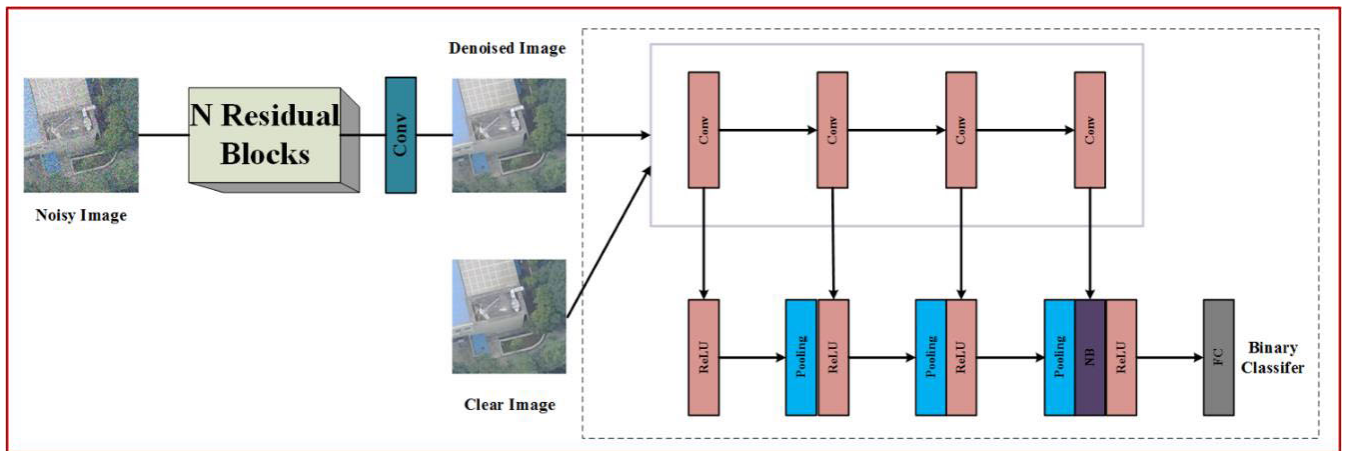


FIGURE 3. Constructing a loss function with a dashed rectangle to learn pixel-class priors.

B. NOISE MODEL

Most classic and practical image denoising models can be applied to tackled the following problems [26]:

$$\tilde{m} = \arg \min_m \frac{1}{2\mu} \|n - m\|^2 + \xi \cdot W(m) \quad (1)$$

The first portion $\frac{1}{2\mu} \|n - m\|^2$ is a data fidelity item having a different noise level ξ , and the second portion $W(m)$ is a regularization term having a generally predefined image prior. ξ is the hyperparameter that balances the two parts. The discriminative denoising model employed in this work is intended to learn the nonlinear mapping function $m = F(n)$ parameterized by P to predict the potentially sharp image m from the noise image n . Therefore, the solution of equation (1) is given by:

$$\tilde{m} = G(n, \mu, \xi, W; P) \quad (2)$$

The hinge to the implementation of this framework is the pre-defined image. This observation prompted us to learn image priors directly from the datasets. In particular,

two data-driven image priors are learned at the feature level and pixel level, respectively. Before constructing a paired training data set, an approximate noise block needs to be extracted from a given noise image. These modules are then applied to train the GAN for noise modeling and noise data generation.

Properly training the GAN to simulate unknown noise is a significant step, as the noise distribution will be better estimated from the noise dominant data. Equation 2 requires a predefined noise level μ , so the trained image denoising model is not flexible enough to handle different noise levels through a GAN network. In order to complete blind image denoising, this study seeks to combine noise level information by previously learning images in the feature-class level space [26]. In particular, this study trains multiple types of discriminators at the fusion feature level and pixel level output of local and global paths to understand different level noise images, as shown in Figs. 2 and 3.

The perceptual discriminator stabilizes and improves the performance of the GAN by embedding a convolutional

portion of the pre-trained deep classification network. Specifically, the extracted features of the output image from the pre-training network are connected to the output of the previous layer and then processed by the learnable convolution operation block. In this study, three-step convolutional blocks were used to implement spatial downsampling, and RnResNet was used for image feature extraction. The final classification is processed from each activation in the feature map. Since the valid receptive field for each activation corresponds to an image block on the original input image data, the discriminator actually predicts each label of each image block. The patch-based discriminator makes it useful for high-frequency modeling in image denoising by limiting the focus of the structure in the local image block.

In order to reduce the influence of the training image background, it is necessary to extract a group of approximate noise blocks from the image with weak background. In this way, noise distribution becomes the main target of model training, so as to make GAN model more accurate. Under the assumption that noise distribution variance is zero, approximate noise data can be obtained by subtracting the average value of relatively smooth modules in the noise image. The smooth module we're talking about here is a very similar area within a component.

Based on the above, a smooth and fast patch search algorithm is utilized in this research. Make u_i and v_j^i denote a size $k \times k$ global patch and a local patch u_i of size $l \times l$, respectively. Each u_i is gotten by scanning the entire noise image with the stride e_g , and each v_j^i is gotten by scanning u_i with the stride e_l . Whether or not u_i is a smooth patch in the algorithm is determined by the difference between the mean and the variance between u_i and v_j^i of each j . In other words, there need first define two constraints [18]

$$\left| \text{Avg} \left(v_j^i \right) - \text{Avg} \left(u_i \right) \right| \leq \phi \cdot \text{Avg} \left(u_i \right) \quad (3)$$

And

$$\left| \text{Var} \left(v_j^i \right) - \text{Var} \left(u_i \right) \right| \leq \varphi \cdot \text{Var} \left(u_i \right) \quad (4)$$

$\text{Avg}(\Delta)$ calculate the mean, and $\text{Var}(\Delta)$ calculate the variance, respectively, $\phi, \varphi \in (0, 1)$. If for each j , two constraints are met, u_i will be considered as a smooth patch and added to the set E .

When $E = \{e_1, e_2, \dots, e_t\}$ is obtained by applying an algorithm to all noise images, the set of approximate noise blocks $R = \{r_1, r_2, \dots, r_t\}$ can be derived by $r_i = e_i - \text{Avg}(e_i)$. The devices used in this research often produce high-resolution images. There are a large number of smooth areas that meet requirements in these images, such as rooms, agriculture and rivers. Therefore, a sufficiently smooth plaque can be found in some limited images, which in other words is that enough noise blocks can be extracted to train the GAN model in the following steps.

When the input noise image is insufficient, the number of noise blocks extracted in the previous section is very limited. In this case, deep RnResNet training using only these

blocks would be unsatisfactory. In order to better promote the denoising performance of the model, one method is to model the noise distribution on these extracted blocks and generate more noise data (in other words, generate any number of samples with more diversity) for RnResNet training.

This study try to minimize the aforementioned loss of the discriminator and maximize it compared to the conversion network. This can be achieved by generating a training strategy for the antagonistic network. As a framework for estimating the generation model, GAN has the ability to learn complex distribution. What's more, GAN can generate noise samples through forward propagation without involving other components. In addition, it can also train data through back-propagation algorithm. In this study, GAN is used to estimate the noise distribution on a group of approximate noise blocks. Due to WGAN [27] can improve the training of GAN and generate high quality samples, as previously described in the relevant work section. Therefore, in this research, WGAN-GP [28] is a evolutionary version of WGAN for learning noise distribution. The loss function in this task is

$$\begin{aligned} \text{Loss}_{\text{GAN}} = & E_{\tilde{m} \sim D_g} [W(\tilde{m})] - E_{m \sim D_r} [W(m)] \\ & + E_{\tilde{m} \sim D_{\tilde{m}}} [\|\nabla_{\tilde{m}} W(\tilde{m})\|_2 - 1]^2 \quad (5) \end{aligned}$$

where D_r is the distribution on R , D_g is the destruction of the generator, and $P_{\tilde{x}}$ is defined as a uniform distribution along the line between the pairs of points sampled from D_r and D_g . The trained image noise model is applied to generate noise samples for increasing R and ultimately to obtain larger Datasets $R' = \{r'_1, r'_2, \dots, r'_w\}$.

C. DEEP RESNET RETROFIT & USING

Many previous studies have proposed to use a large number of pairs of datasets which contained different noise levels to train CNN to solve the problem of image noise reduction, and achieved significant results. CNN has the power of a network architecture that implicitly learns potential noise models from paired training datasets, thereby relaxing the dependence on image prior knowledge of human knowledge [29]. Therefore, the ResNet [29] module is used in our framework.

The prediction of clean images can be gotten by $y_i - R(y_i; \Theta)$ (as shown in Fig. 4). The filter number of the last block is equal to the output channel number. Each other block contains 64 filters. In order to train ResNet, you need to first establish a paired training data set. The set V' obtained by noise modeling using GAN, another set of images is divided into small blocks of size $k \times k$, forming a set $X = \{x_1, x_2, \dots, x_e\}$. Randomly add the noise block in V' to the patch in X to get $Y = \{y_1, y_2, \dots, y_f\}$, where $y_1 = x_j + v'_k$. The set X and Y form a paired training datasets $\{X, Y\}$. In fact, the data set was built during the denoising network training. In each training epoch, the combination of x_j and v'_k is changed and a new dataset $\{X, Y\}$ is gotten, which results data increase in next step [30].

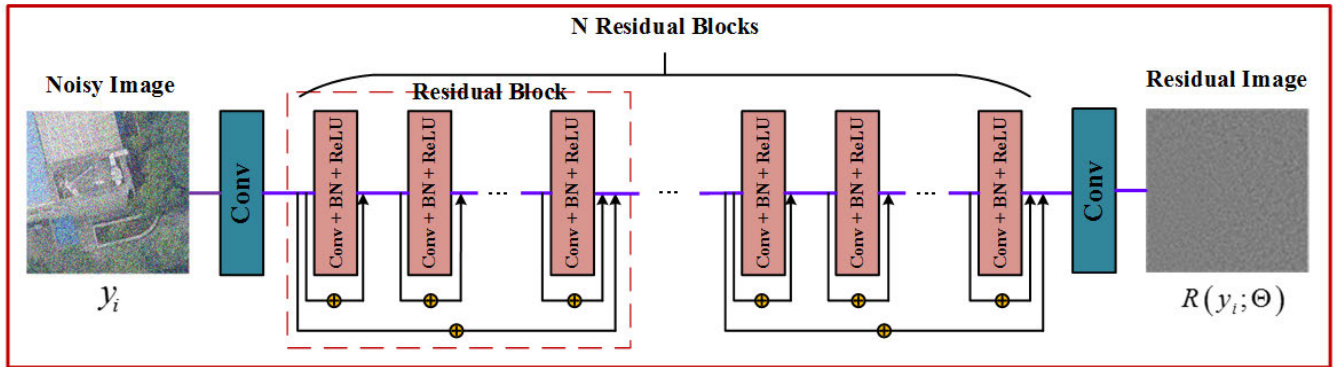


FIGURE 4. Convolutional network architecture of the DnResNet. The input on the left is the noise image dataset y_i and the output result on the right is $R(y_i; \Theta)$ which is the difference between the input and the potentially clean image.

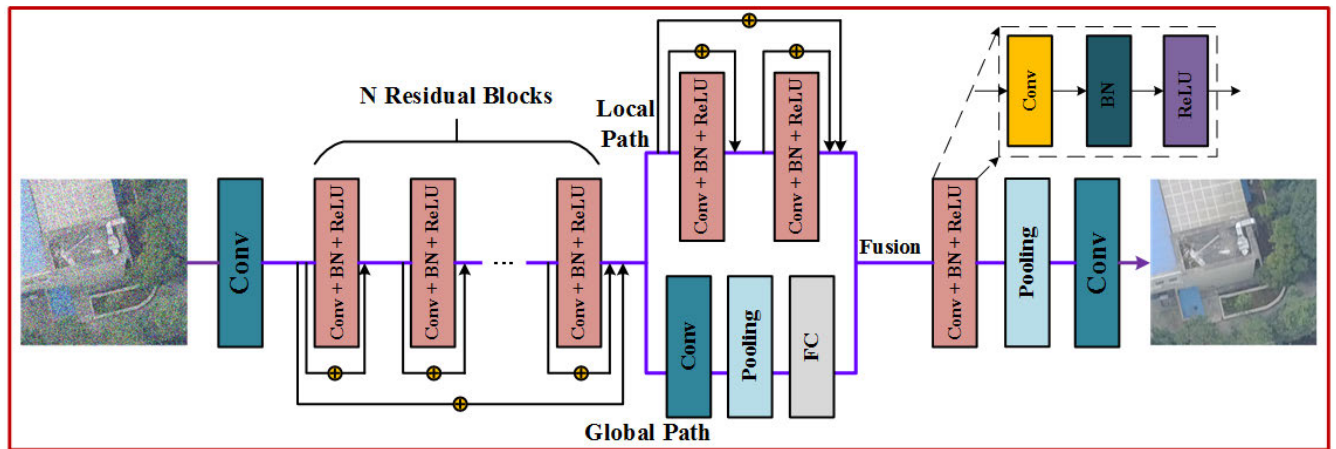


FIGURE 5. Convolutional network architecture of the image denoising network proposed by this study. The input noise image dataset is first processed by the N Residual Blocks (a depth residual network) to calculate low feature-class and pixel-class level features that are then divided into two paths to learn other local and global features. Our image denoising model then fuses the two paths to produce the final output.

When the paired noise-clean training dataset is established, DnResNet can be trained to eventually denoise (as shown in Fig. 5). In this experiment, a network structure similar to DnCNN was employed. DnResNet is considered to be a single residual block which predicts the residual image data, in other words is that predicts the difference between the input noise image and the potentially clean image data [31], [32]. The loss function to be minimized is set as following

$$Loss_{DnResNet}(\Theta) = \frac{1}{2\theta} \sum_{i=1}^{\theta} \left\| R^2(y_i; \Theta) - (y_i - x_i)^2 \right\|_F^{\frac{1}{2}} \quad (6)$$

where Θ is the parameter of the DnResNet, θ is the size of the training image dataset, y_i is the noisy image data, and x_i is the base image data. Batch normalization, ReLU and residual learning strategies are also used to improve the training of deep networks.

The image noise reduction network in this paper consists of three parts: a pile of residual blocks (as shown in Fig. 4 or 5) for extracting low-level features of the input image; and two asymmetric paths for extracting local and global features, respectively. Then our architecture fuses these two paths to produce the final output.

The input noise image is first processed by a 16-layer residual network with skip connections to extract low-level features (Fig. 4). The “pre-activated” residual block is used because it is easier to train and promote than the original ResNet. For all residual blocks, use kernel size and zero rewrite to keep the size of the input space. This study also keep the number of features in all remaining blocks at 32. In addition, a skip join is added between the input feature and the output of the last residual block. As a result, large space support can be utilized to extract complex patterns [33]. The coding features are further processed by two asymmetric networks for local and global feature extraction. The local path is fully convolved and consists of two residual blocks, as shown in Fig. 4. It is designed to learn local features while preserving spatial information. The remaining connections make it easy to learn the same function, considering that the output image shares many structures with the input image, which is an attractive feature for the conversion network. The global path uses two fully connected layers to learn global features. Each fully connected layer is followed by a ReLU layer as an activation function. The global average merge layer is used to ensure that our model can handle images of any resolution [34], [35]. Finally, the global information is

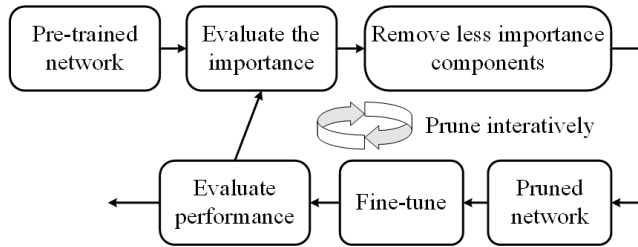


FIGURE 6. Representative process of incremental model pruning. There are four iterative steps: (1) assessing the importance of each component in the pre-trained depth model; (2) removing components that are less important to model inference; and (3) fine-tuning the trim model to compensate for potential temporary performance degradation; (4) Evaluate the fine-tuning model to determine if the pruning model is suitable for deployment. Make sure that the incremental pruning strategy is preferred to prevent over-pruning.

summarized as a fixed dimension vector and used to normalize the local features produced by the local path. The local and global features are then fused into a common set of features that are fed to the convolutional layer to produce an output [36].

D. MODEL PRUNING

When deploying a depth model on a resource-constrained device, model compression is a useful tool for researchers to re-adjust the resource consumption required by the depth model. Existing model compression methods mainly include model pruning [37], knowledge distillation [38], [39], parameter quantification and dynamic calculation. In this section, we discuss model pruning methods in detail.

The representative process of incremental model pruning is shown in Fig. 6. The components removed from the deep model in the model pruning method can be separate neural connections or network structures. The weight trim method trims less important connections with less weight. It is conceptually easy to understand, but due to the generated irregular network architecture, it is difficult to store the pruned model and speed it up. Technically, weight trimming may not be suitable for practical applications unless a dedicated software library or dedicated hardware is designed to support the trim model. Unlike weight pruning, structured pruning is more likely to produce a regular and manageable network architecture. In order to obtain the structural importance of structured pruning, the researchers used sparse training [40], using structured sparsity regularization, including structured sparsity learning and channel scaling factor sparsity. Liu et al. proposed a simple but effective channel clipping method called network weight loss [40]. They directly use the scaling factor in the bulk normalization (BN) layer as the channel mode scaling factor, and perform L1 regularization training networks on these scaling factors to obtain channel mode sparsity. Channel pruning is a coarse-grained but effective method and, more importantly, it is convenient to implement a pruning model without dedicated hardware or software requirements [41]. They applied a network slimming method to pruning CNN-based image classifiers and significantly

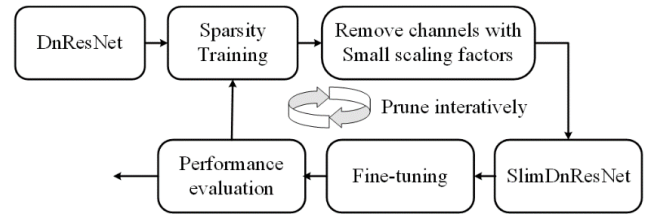


FIGURE 7. Iterative process of efficient depth image denoiser learning through SlimRGBD sparse training and channel pruning.

reduced model size and computational operations [40]. In this paper, we follow Liu's work and extend it to a coarse-grained neural structure search method to find effective depth image noise reducers using the pruned RnResNet.

Manually designing the network architecture of the depth image noise reducer does not guarantee that each component plays an important role in forward reasoning. It is recommended to learn an effective depth image noise reducer by performing channel clipping on the convolutional layer. Specifically, the goal of this paper is to search for a more compact and efficient convolutional channel configuration to help reduce trainable parameters and FLOP. To this end, channel trimming is applied in SlimRGBD to obtain SlimRGBD by following the procedure shown in Fig. 7.

After sparsity training, this study introduces a global threshold ξ to determine if the feature channel is to be trimmed. The global threshold ξ is set to the n th percentile of all n to control the trim ratio. In addition, this study introduced local security thresholds to prevent excessive convolution of the convolutional layer and maintain the integrity of the network connection. The local security threshold is set in a hierarchical manner to the k th percentile of all k s in a particular layer. This study trims the feature channel whose scale factor is less than the minimum of the minimum. In SlimRGBD, several special connections between layers need to be carefully handled. During the pruning process, this study discards the maxpool layer and the upsampled layer directly because they are independent of the channel number. Initially, this study constructed a trim mask for all convolutional layers based on global thresholds and local safety threshold constructs. For path layers, this study connects the clipping masks of their incoming layers in order and uses the connection mask as their clipping mask. The shortcut layer in SlimRGBD is similar to the residual learning in ResNet. Therefore, all layers connected to the shortcut layer need to have the same channel number. To match the feature channels of each layer connected by the shortcut layer, this study traverses the trim masks of all the connection layers and performs an OR operation on these trim masks to generate the final trim mask for these tie layers. After trimming the channel, it is recommended to perform a fine-tuning operation on the trim model to compensate for potential temporary degradation. In fine-grained object inspection tasks, detection performance is often sensitive to channel pruning. Therefore, fine-tuning is important to make the pruning model recover from a potential reduction in performance.

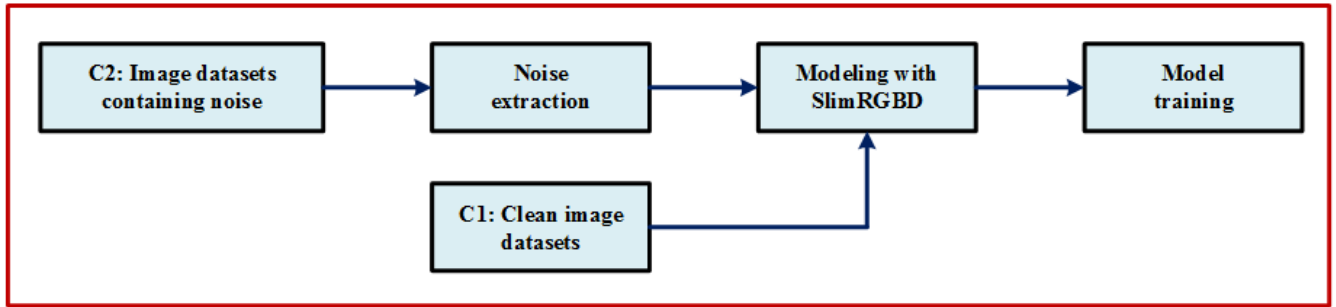


FIGURE 8. Experimental process.

E. SPARSE TRAINING

The channel mode sparsity of the depth model helps channel trimming and describes the number of less important channels that may be deleted later. To facilitate channel pruning, we assign a scale factor to each channel, where the absolute value of the scale factor represents the channel importance. Specifically, in addition to detecting the header, the BN layer after each convolutional layer in SlimRGBD accelerates convergence and improves generalization [42]. The BN layer uses small batch static to normalize the convolution feature, which is formulated as equation (7).

$$y = \gamma \times \frac{x - \bar{x}}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{7}$$

\bar{x} and σ^2 are the mean and variance of the input elements in the mini batch, and γ and β represent the scale factors and deviations that can be trained. Of course, we directly use the trainable scale factor in the BN layer as an indicator of channel importance. In order to effectively distinguish between important channels and unimportant channels, we perform channel mode sparsity training by implementing L1 regularization on γ [43]. The training goal of sparse training is given by equation (8).

$$L = loss_{GCBD} + \alpha \sum_{\gamma \in \Gamma} f(\gamma) \tag{8}$$

where $f(\gamma) = |\gamma|$ represents the L1 norm and α represents the penalty factor that balances the two loss terms.

IV. EXPERIMENT AND ANALYSIS

A. DATASETS

Train the model on the LY_Datasets, which is a remote sensing image dataset. In the present study, all of them were randomly tailored to 64×64 image patches for training. The input noise image is added by the noise level σ in the [10, 80] range, and the corresponding sharp image is used as the GroundTruth.

B. EXPERIMENT

For the proposed SlimRGBD in this research, a set of clean image datas is applied to construct a paired noise-clean training dataset with noise image data generated by the GAN network. In order to simulate the environment for actually

processing a large image, noise data is added in another set of high resolution clean image dataset to form a noise image dataset for SlimRGBD in the estimate of the synthesized data.

In the image noise extraction procedure, the parameter: d, h, s_g, s_l and γ is set to 64, 16, 32, 16, 0.1 and 0.25, severally. For noise GAN modeling, this article generally follows the parameter settings in DCGAN [44], [45]. For RnResNet, it trains 50 periods with an initial learning rate of 0.0001 and an Adam [46] optimizer (as shown in Fig. 8, the experimental flow of the study is shown. As shown in Fig. 9, this study also compares the process of using SGD [47] training).

The competition methods compared in this study include BM3D [48], EPLL [49], NCSR [50], WNNM [51], Multi-scale [50], DnCNN [17] and the proposed SlimRGBD. In particular, in order to reveal the limitations of the method based on discriminative learning in dealing with blind denoising problems, a blind model of DnResNet available for Gaussian image denoising is employed in the evaluation. Specifically, the DnCNN is trained with accurate Gaussian noise image data from different class levels, which achieves the prior art blind Gaussian denoising results.

The SlimRGBD in this paper uses a 3×3 core for the transform network and all convolutional layers in both discriminator networks. After each convolution layer is constant followed by a batch normalization layer to stabilize and accelerate deep network training. The model was trained using Adam optimizer to achieve a random optimization of 64 in 45 epoches with a convergence time of approximately 43 hours. The initial learning rate is 0.0001, and the introduced cosine shape learning rate table smoothes the initial learning rate. Implement a deep learning framework based on TensorFlow and a single GTX 1080Ti GPU.

C. INTEGRATED NOISE ASSESSMENT

In this segment, different types of zero-avg synthesized noise image data are generated and added to LY_Datasets to evaluate all competing methods. In this evaluation, in addition to DnCNN and SlimRGBD, other methods have actual noise levels (ie, standard deviation σ).

This section evaluates the SlimRGBD method for synthetic and actual data. Several representative methods were compared. Four sections of experiments were carried out: (1) evaluating the accuracy of noise GAN modeling, and comparing

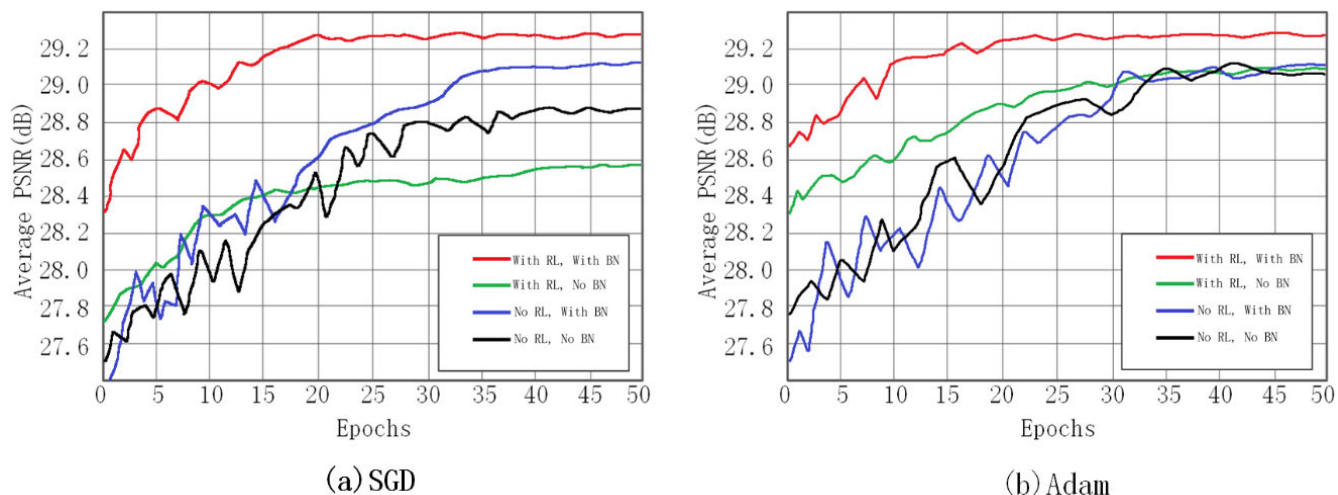


FIGURE 9. Under two gradient-based optimization algorithms ((a) SGD, (b) Adam), four specific image denoising models are trained in different combinations of residual learning (RL) and batch normalization (BN). Gaussian denoising results are shown in the two figures (with a noise level of 20). The results were evaluated on 80 natural images from LY_Datasets.

TABLE 1. Results of PSNR (DB) for all comparison methods on LY_datasets in the synthetic noise denoising environment.

Gaussian Noise								
Mode	Non-Blind				Blind			
Method	BM3D	EPLL	NCSR	WNNM	Multiscale	DnCNN	G CBD	SlimRGBD
$\sigma = 20$	27.03	28.01	27.79	28.66	26.52	28.84	28.73	29.27
$\sigma = 35$	24.88	24.95	24.90	25.17	24.60	25.36	25.34	25.55
Mixture Noise								
Mode	Non-Blind				Blind			
Method	BM3D	EPLL	NCSR	WNNM	Multiscale	DnCNN	G CBD	SlimRGBD
$s = 20$	36.09	36.06	36.05	36.02	34.78	36.57	38.01	38.29
$s = 35$	33.48	33.36	33.59	33.32	33.28	33.29	35.77	35.98

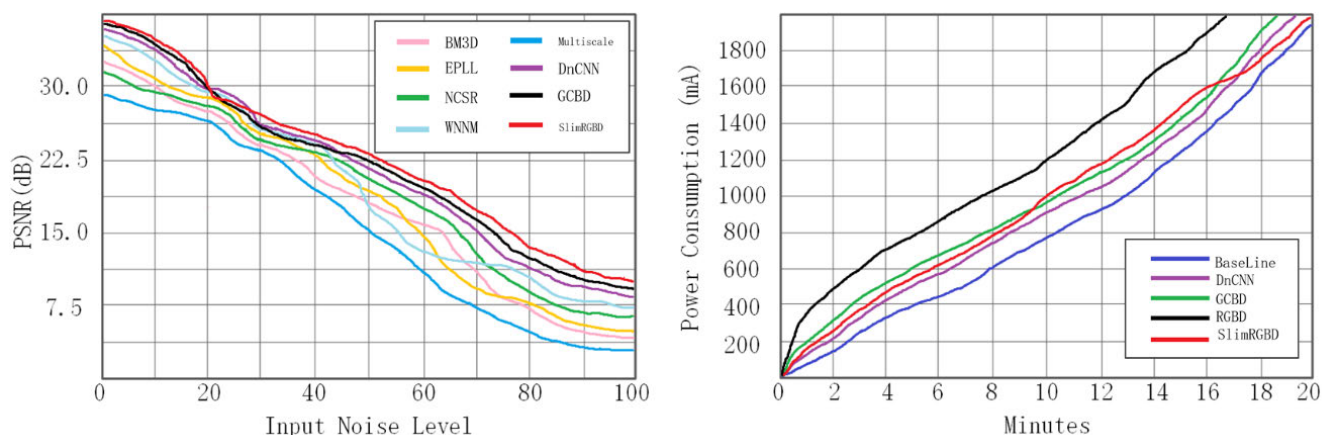


FIGURE 10. Left: Noise sensitivity curve for the SlimRGBD model of noise level training. The average PSNR results at different input noise levels were evaluated on LY_Datasets. Right: Based on the UAV power consumption baseline, Loading 4 kinds of intelligent noise reduction algorithms the power consumption. It can be seen that the RGBD noise reducer without pruning consumes a large amount of power. The SlimRGBD algorithm proposed in this study consumes less power than DnCNN in some cases.

SlimRGBD with the prior art denoising method, especially based on the discriminative learning method DnCNN, in the Gaussian blind denoising task; (2) In order to show that SlimRGBD can process more complex noises than Gaussian

noise, this experiment uses mixed noise to evaluate; (3) discusses the selection of noise modeling methods, shows noise samples, and shows that the reason which GAN is selected and not other traditional methods is selected, such as GMM.

TABLE 2. Time spent testing various noise reduction methods on DJI phantom 4 pro V2.0 (in seconds).

	BM3D	EPLL	NCSR	WNNM	Multiscale	DnCNN	GCBD	SlimRGBD
Time (s)	0.015	0.016	0.012	0.010	0.008	0.024	0.026	0.018



Test 1



Test 2

FIGURE 11. The first set of test samples. Test 1&2 is the original two aerial image of drones randomly extracted from the LY Datasets. The shaded squares in the figure are the parts we will show in Fig. 12.

Test 3



Test 4

FIGURE 12. The second set of test samples. Test 3&4 is the original aerial image of two drones randomly extracted from the LY Datasets data set. The shaded squares in the figure are the parts we will show in Fig. 13.

A large number of experiments have proved the superiority of SlimRGBD in image blind denoising.

Since Gaussian noise is one of the widely studied noises, it is important to perform blind Gaussian denoising experiments. Table 1 shows the different results for all comparison methods. Although no noise information is provided, SlimRGBD is still superior to BM3D, EPLL, WNNM and Multiscale. In particular, SlimRGBD and DnCNN achieve comparable results. This is impressive because DnCNN uses accurate data for training, while SlimRGBD uses the approximate data generated by GAN for training. This experiment demonstrates the accuracy of noise modeling by using GAN.

In addition to Gaussian noise (which a noise of Hybrid Noise), we further evaluated the performance of other several methods in complex noise denoising environment. The mixed noise used in the experiment includes 10% uniform noise, 20% Gaussian noise, and 70% Gaussian noise. Table 1 shows the quantitative results. In this research, SlimRGBD also performs better than EPLL, BM3D, Multiscale and WNNM, which further demonstrates the superiority of SlimRGBD in blind denoising problems. In particular, DnCNN does not perform well due to the paired training data set is not available. In contrast, the proposed SlimRGBD uses GAN to estimate the image noise distribution of the noise image

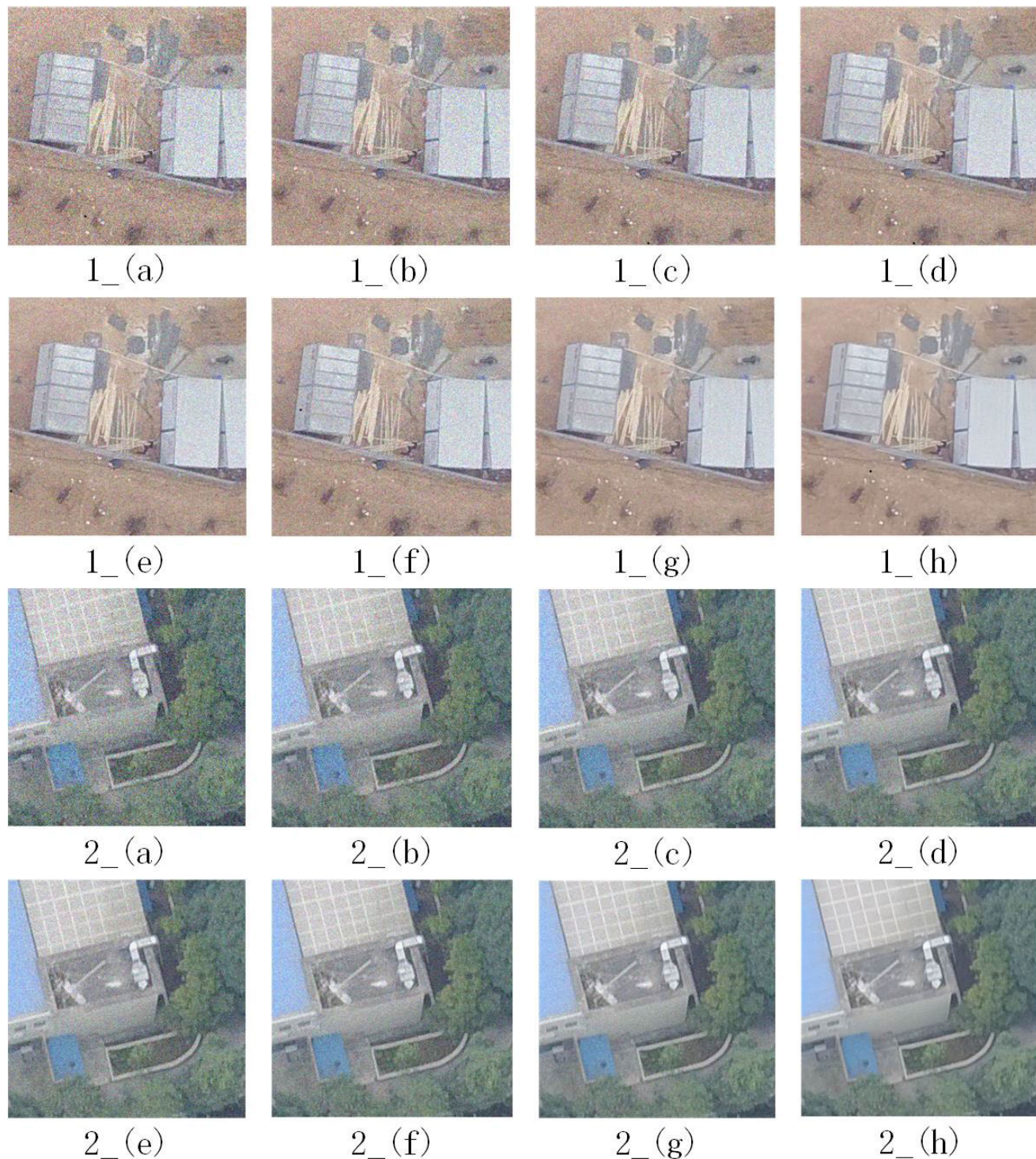


FIGURE 13. Comparison of housing areas when evaluating real noise denoising. The contrast areas in the figure are from the shaded squares in Fig. 10, respectively. Here, zoom in for a better visual comparison. The serial numbers in the figure: (a) (b) (c) (d) (e) (f) (g) (h) represent (a) original image; (b) EPLL; (c) NCSR; (d) WNNM; (e) Multiscale; (f) DnCNN; (g) GCBD; (h) results graph after SlimRGBD processing. As can be seen from their comparison results, SlimRGBD is a very competitive image denoising method.

and solves the problem of lack of training image data, thereby achieving significant denoising effects, as shown in Fig. 10 (left). Fig. 10 (right) is based on the UAV power consumption baseline, Loading 4 kinds of intelligent noise

reduction algorithms, the power consumption. It can be seen that the RGBD noise reducer without pruning consumes a large amount of power. The SlimRGBD algorithm proposed in this study consumes less power than DnCNN in

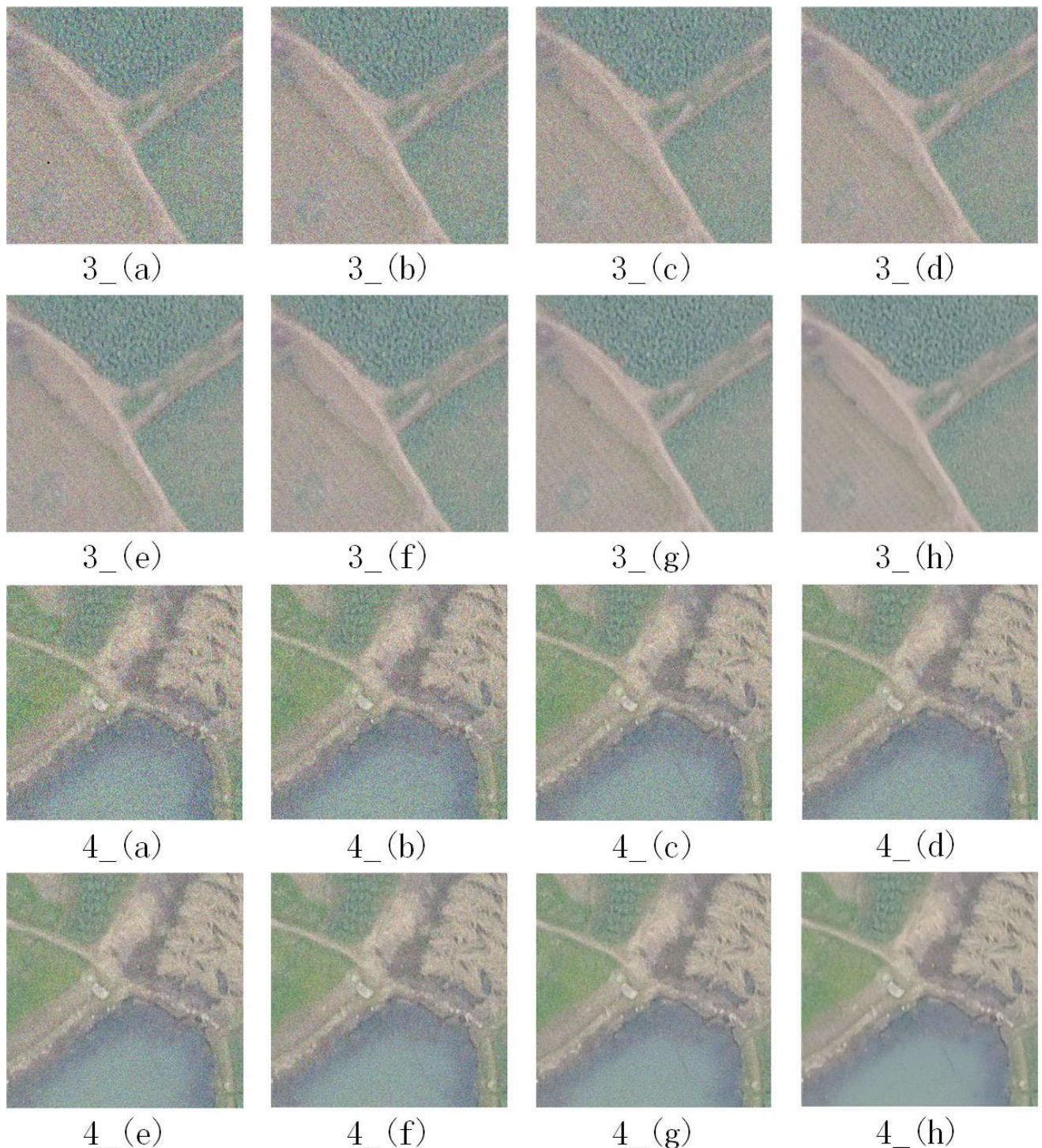


FIGURE 14. Comparison of agricultural and forestry areas when assessing true noise denoising. The contrast areas in the figure are from the shaded squares in Fig. 12, respectively. Here, zoom in for a better visual comparison. The serial numbers in the figure: (a) (b) (c) (d) (e) (f) (g) (h) represent (a) original image; (b) EPLL; (c) NCSR; (d) WNNM; (e) Multiscale; (f) DnCNN; (g) GCBD; (h) Results graph after SlimRGBD processing. As can be seen from their comparison results, SlimRGBD is a very competitive image denoising method.

some cases. For the above cases (Mixed noise or Gaussian noise), the proposed method in this research performs well. Specifically, the deviation value of the GAN from the distribution of the extracted noise block learning to the GroundTruth distribution is about 0.25% of the average value

and 1.14% of the standard deviation. Time spent testing various noise reduction methods on DJI Phantom 4 Pro V2.0 is as Table 2. It can be seen that although our method uses more time consuming than some traditional methods, it is the best among all neural network methods.

D. SLIMRGBD PERFORMANCE AND APPLICATION ANALYSIS

The most vital part of the proposed system is noise modeling, which involves extracting noise blocks and learning the noise distribution through the GAN. In this section, it will first study the effects of the first step and then discuss the accuracy of noise modeling which used GAN.

To check the effectiveness of GAN noise modeling, Effect of Noise Modeling Using GAN was been used, which accurate synthetic noise data is used as input to train SLIMRGBD system. In addition, SLIMRGBD can well learn the noise distribution and generate good samples when dealing with complex real-world noise (see Figs. 11, 12, 13, 14). All these facts show that using GAN to simulate noise may be accurate.

V. CONCLUSION AND FUTURE WORK

This study attempts to improve the performance of image blind denoising by using a deep learning-based approach without paired training data. The proposed SlimRGBD can improve blind denoising performance. The GAN is used to learn the noise distribution and construct a paired training data set to train the DnResNet for denoising. A large number of experiments have proved the superiority of our method. In the future work, we will integrate our work into more embedded platforms to improve the image noise reduction performance of other devices.

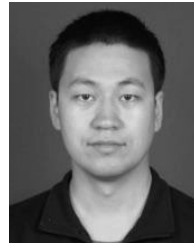
ACKNOWLEDGMENT

All of authors in this paper would like to thanks for all anonymous reviewers for their very insightful comments and constructive suggestions to polish this paper in high quality.

REFERENCES

- [1] S. F. Di Gennaro, A. Matese, B. Gioli, P. Toscano, A. Zaldei, A. Palliotti, and L. Genesio, "Multisensor approach to assess vineyard thermal dynamics combining high-resolution unmanned aerial vehicle (UAV) remote sensing and wireless sensor network (WSN) proximal sensing," *Sci. Horticulture*, vol. 221, pp. 83–87, Jul. 2017.
- [2] J. Cao, W. Leng, K. Liu, L. Liu, Z. He, and Y. Zhu, "Object-based mangrove species classification using unmanned aerial vehicle hyperspectral images and digital surface models," *Remote Sens.*, vol. 10, no. 2, p. 89, Jan. 2018.
- [3] N. Bagheri, "Development of a high-resolution aerial remote-sensing system for precision agriculture," *Int. J. Remote Sens.*, vol. 38, nos. 8–10, pp. 2053–2065, May 2017.
- [4] P. Thenkabail, "Remote sensing systems-platforms and sensors: Aerial, satellite, UAV, optical, radar, and LiDAR," in *Remote Sensing Handbook—Three Volume Set*. Boca Raton, FL, USA: CRC Press, 2018, pp. 37–92.
- [5] Z. Y. Sun, Y. Q. Chen, L. Yang, G. L. Tang, S. X. Yuan, and Z. W. Lin, "Small unmanned aerial vehicles for low-altitude remote sensing and its application progress in ecology," *Chin. J. Appl. Ecol.*, vol. 28, no. 2, pp. 528–536, 2017.
- [6] W. Xiang, W. Xinxin, Z. Jianhua, F. Jianchao, S. Xiu, and Z. Dejun, "Monitoring the thermal discharge of hongyanhe nuclear power plant with aerial remote sensing technology using a UAV platform," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017.
- [7] H. Xiang and L. Tian, "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV)," *Biosyst. Eng.*, vol. 108, no. 2, pp. 174–190, Feb. 2011.
- [8] M. E. Helou and S. Susstrunk, "Blind universal Bayesian image denoising with Gaussian noise level learning," 2019, *arXiv:1907.03029*. [Online]. Available: <https://arxiv.org/abs/1907.03029>
- [9] S. Pierre, J. Armand, G. Remi, G. Benjamin, and J. Herve, "And the bit Goes down revisiting the quantization," 2019, *arXiv:1907.05686v1*. [Online]. Available: <https://arxiv.org/abs/1907.05686v1>
- [10] P. Zhang, Y. Zhong, and X. Li, "SlimYOLOv3: Narrower, faster and better for real-time UAV applications," 2019, *arXiv:1907.11093*. [Online]. Available: <https://arxiv.org/abs/1907.11093>
- [11] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [12] F. Zhu, G. Chen, J. Hao, and P.-A. Heng, "Blind image denoising via dependent Dirichlet process tree," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1518–1531, Aug. 2017.
- [13] 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.
- [14] B. Benoit and V. Christophe De, "AF-DCGAN: Amplitude feature deep convolutional GAN for fingerprint construction in indoor localization system," 2019, *arXiv:1906.00270v1*. [Online]. Available: <https://arxiv.org/abs/1906.00270v1>
- [15] V. Iglovikov and A. Shvets, "TernausNet: U-Net with VGG11 encoder pre-trained on ImageNet for image segmentation," 2018, *arXiv:1801.05746*. [Online]. Available: <https://arxiv.org/abs/1801.05746>
- [16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent.*, 2015.
- [17] 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.
- [18] 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.
- [19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, and S. Ozair, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [20] P. Andreini, S. Bonechi, M. Bianchini, A. Mecocci, F. Scarselli, and A. Sodi, "A two stage GAN for high resolution retinal image generation and segmentation," 2019, *arXiv:1907.12296*. [Online]. Available: <https://arxiv.org/abs/1907.12296>
- [21] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," 2018, *arXiv:1801.02610*. [Online]. Available: <https://arxiv.org/abs/1801.02610>
- [22] X. Xue, C. Wu, Z. Sun, Y. Wu, and N. Xiong, "Vegetation greening for winter oblique photography using cycle-consistency adversarial networks," *Symmetry*, vol. 10, no. 7, p. 294, Jul. 2018.
- [23] G. Alain, Y. Bengio, L. Yao, J. Yosinski, É. Thibodeau-Laufer, S. Zhang, and P. Vincent, "GSNs: Generative stochastic networks," *Inf. Inference*, vol. 5, no. 2, pp. 210–249, Jun. 2016.
- [24] S. Xi, C. Wu, and L. Jiang, "Super resolution reconstruction algorithm of video image based on deep self encoding learning," *Multimedia Tools Appl.*, vol. 78, no. 4, pp. 4545–4562, Feb. 2019.
- [25] V. Edupuganti, M. Mardani, J. Cheng, S. Vasanaawala, and J. Pauly, "VAE-GANs for probabilistic compressive image recovery: Uncertainty analysis," *Tech. Rep.*, 2019.
- [26] X. Hou, H. Luo, J. Liu, B. Xu, K. Sun, and Y. Gong, "Learning deep image priors for blind image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019.
- [27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: <https://arxiv.org/abs/1701.07875>
- [28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017.
- [29] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 512–519.
- [30] Q. Hu, C. Wu, Y. Wu, and N. Xiong, "UAV image high fidelity compression algorithm based on generative adversarial networks under complex disaster conditions," *IEEE Access*, vol. 7, pp. 91980–91991, 2019.
- [31] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the ResNet model for visual recognition," *Pattern Recognit.*, vol. 90, pp. 119–133, Jun. 2019.
- [32] S. A. Bigdeli, M. Zwicker, P. Favaro, and M. Jin, "Deep mean-shift priors for image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 763–772.

- [33] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1671–1681.
- [34] D. Ren, W. Zuo, D. Zhang, L. Zhang, and M.-H. Yang, "Simultaneous fidelity and regularization learning for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
- [35] M. S. Hosseini and K. N. Plataniotis, "Convolutional deblurring for natural imaging," *IEEE Trans. Image Process.*, vol. 29, pp. 250–264, 2020.
- [36] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016, *arXiv:1612.00215*. [Online]. Available: <https://arxiv.org/abs/1612.00215>
- [37] C. Zhao, B. Ni, J. Zhang, Q. Zhao, W. Zhang, and Q. Tian, "Variational convolutional neural network pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2780–2789.
- [38] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017.
- [39] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017.
- [40] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature Commun.*, vol. 9, p. 2383, Jun. 2018.
- [41] M. Zohrer, R. Peharz, and F. Pernkopf, "Representation learning for single-channel source separation and bandwidth extension," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 12, pp. 2398–2409, Dec. 2015.
- [42] F. Pozas-Flores, R. Carmona-Galan, and A. Rodriguez-Vazquez, "Simplified state update calculation for fast and accurate digital emulation of CNN dynamics," in *Proc. Int. Workshop Cellular Nanosc. Netw. Appl.*, 2010.
- [43] M. C. Muelly and L. Peng, "Spotting brain bleeding after sparse training," *Nature Biomed. Eng.*, vol. 3, no. 3, pp. 161–162, Mar. 2019.
- [44] P. L. Suarez, A. D. Sappa, and B. X. Vintimilla, "Infrared image colorization based on a triplet DCGAN architecture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017.
- [45] Q. Li, H. Qu, Z. Liu, N. Zhou, W. Sun, and J. Li, "AF-DCGAN: Amplitude feature deep convolutional gan for fingerprint construction in indoor localization system," 2018, *arXiv:1804.05347*. [Online]. Available: <https://arxiv.org/abs/1804.05347>
- [46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [47] S. Theodoridis, "Stochastic gradient descent," *Mach. Learn.*, pp. 161–231, 2015.
- [48] M. V. Subramanyam and M. N. G. Prasad, "A modified BM3D algorithm for SAR image despeckling," *Procedia Comput. Sci.*, vol. 70, pp. 69–75, Jan. 2015.
- [49] L. Bin and L. Yuanyuan, "EPLL based natural image restoration using spatially constrained Gaussian mixture model," *Chin. J. Quantum Electron.*, vol. 32, no. 4, pp. 391–398, 2015.
- [50] V. Sa-ing, P. Vorasayan, N. C. Suwanwela, S. Auethavekiat, and C. Chirungrueng, "Multiscale adaptive regularisation Savitzky–Golay method for speckle noise reduction in ultrasound images," *IET Image Process.*, vol. 12, no. 1, pp. 105–112, Jan. 2018.
- [51] Z. Jie and W. Peipei, "SAR image denoising based on NSST with WNNM and KAD," *J. Zhengzhou Univ. (Natural Sci. Ed.)*, to be published.



BOBO JU is currently pursuing the master's degree in computer science and technology with the University of Shanghai for Science and Technology. He is also engaged in big data and artificial intelligence research under the guidance of Prof. C. Wu.



YAN WU received the Ph.D. degree from Southern Illinois University Carbondale, with concentrations in environmental chemistry and ecotoxicology. He is currently a Postdoctoral Associate with the O'Neill School of Public and Environmental Affairs, Indiana University Bloomington. His research involves elucidations of environmental fate of contaminants using chemical and computational techniques, predictions of their associated effects on wildlife and public health, and data processing and analysis in environmental related fields.



NAIKUE XIONG (Senior Member, IEEE) received the Ph.D. degree in sensor system engineering from Wuhan University, and the Ph.D. degree in dependable sensor networks from the Japan Advanced Institute of Science and Technology. Before he attended Tianjin University, China, he was with Northeastern State University, Georgia State University, Wentworth Technology Institution, and Colorado Technical University (Full Professor about five years) about ten years. He is currently a Professor with the College of Intelligence and Computing, Tianjin University. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory. He has published over 300 international journal articles and over 100 international conference papers. Some of his works were published in IEEE JSAC, IEEE or ACM TRANSACTIONS, ACM Sigcomm Workshop, IEEE INFOCOM, ICDCS, and IPDPS. He is a Senior Member of the IEEE Computer Society. He received the Best Paper Award in the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08) and the Best Student Paper Award in the 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009). He has been the General Chair, the Program Chair, the Publicity Chair, a PC Member, and a OC Member of over 100 international conferences, and a reviewer of about 100 international journals, including IEEE JSAC, IEEE SMC (Park: A/B/C), the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON MOBILE COMPUTING, and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He is serving as the Editor-in-Chief and an Associate Editor or an Editor Member for over ten international journals, including an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS: SYSTEMS, *Information Science*, the Editor-in-Chief for the *Journal of Internet Technology* (JIT) and the *Journal of Parallel and Cloud Computing* (PCC), and a Guest Editor for over ten international journals, including *Sensors Journal*, WINET, and MONET. He is also the Chair of Trusted Cloud Computing Task Force, the IEEE Computational Intelligence Society (CIS), and the Industry System Applications Technical Committee.

...



CHUNXUE WU (Member, IEEE) received the Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Beijing, China, in 2006. He is currently a Professor with the Computer Science and Engineering and Software Engineering Division, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include, wireless sensor networks, distributed and embedded systems, wireless and mobile systems, networked control systems.