IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

SPECIAL SECTION ON RECENT ADVANCES IN VIDEO CODING AND SECURITY

Received June 11, 2019, accepted July 2, 2019, date of publication July 9, 2019, date of current version July 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927809

UAV Image High Fidelity Compression Algorithm Based on Generative Adversarial Networks Under Complex Disaster Conditions

QIUHONG HU[1], CHUNXUE WU[1], YAN WU[2], AND NAIXUE XIONG[3,4]

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]School of Public and Environmental Affairs, Indiana University Bloomington, Bloomington IN 46202-5199, USA
[3]Department of Computer Science and Technology, Tianjin University, Tianjin 300072, China
[4]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

Corresponding authors: Chunxue Wu (wcx@usst.edu.cn) and Naixue Xiong (xiongnaixue@gmail.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0810204 and Grant 2018YFB17026, 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 Laboratory of Modern Optical System.

ABSTRACT This paper proposes an improved image high fidelity compression algorithm based on the generative adversarial networks (GANs) to deal with the problem that the UAV image has a large amount of data which is not conducive to post-processing. By adding an encoder in front of the generator, the disaster area image transmitted by UAV is compressed to meet the requirements of the generator. After the compressed image is trained together with the real image through the discriminator, the quality of the compressed image is constantly improved. This image compression algorithm can fully synthesize the codes of non-major areas such as trees and rivers in the image, and try to retain the codes of important areas such as houses and roads. The experimental results show that the proposed compression method in this paper has a higher compression ratio than the traditional compression method for the disaster area image, and can obtain images with strong sense of hierarchy.

INDEX TERMS Complex disaster, convolutional neural networks, discriminator, generative adversarial networks, generator, high fidelity, image compression.

I. INTRODUCTION

In recent years, a variety of natural and man-made disasters have occurred frequently, seriously affecting people's lives in disaster areas. Although the damage caused by disasters can be understood through various channels, people still lack a clear understanding of the overall impact of disasters [1]. With the continuous development of uav technology, it has been applied in many fields such as disaster assessment, environmental detection, traffic management and aerial photography recording [2]. The purpose of applying uav to complex disaster relief is to replace the traditional and inefficient manual rescue, and complete various tasks with high efficiency and high reliability. The first priority for disaster relief is to transmit high-definition disaster images, and eliminating redundant data is the key to high-quality transmission. In this paper, a high fidelity image compression method is proposed to reduce the pressure of storage device capacity, and facilitate the later image processing [3].

Compared with other methods, video and image information give people an intuitive feeling, and people mainly obtain external information through them. However, video and image contain a large number of invalid data, mainly involving time redundancy, space redundancy and coding redundancy [4]. Image compression is the basis of processing and transmission. If the image is not compressed in the early stage, it will increase the difficulty of image stitching in the later stage and affect the transmission efficiency. This may hinder people from using the effective data of video and image [5]. How to transmit clear high-fidelity images with limited bandwidth and get high quality compressed data has become the focus of research. Image compression has become a frontier science in the field of computer vision [6].

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

Compression technology can be roughly divided into three stages, each of which produces a number of creative results. The first compression coding stage is mainly used to remove redundancy, such as PCM [7] and transformation coding method [8]. The emergence of arithmetic coding [9] marks the beginning of the second generation of image compression coding, followed by dictionary coding [10] and lossless compression algorithm. The third generation compression technology mainly includes fractal image, wavelet transform [11] and other coding algorithms, and has been developing till now. The image recognition and synthesis method are used to compress the data, mainly including the pixel coding, prediction coding [12] and other main technologies. In recent years, deep learning [13]–[16] has been widely used in data analysis [17], image recognition [18], speech processing [19] and other fields. Compression based on neural network structure is a hot topic in current research. The latest image compression algorithm combines the neural network structure. The compressed size is only about half of the size of the traditional algorithm compression, and the visual quality is not affected [20]. Convolutional neural network is a common network structure in the field of image compression [21].

The compression of disaster image should consider not only the details of each part, but also ensure that the image has high fidelity [22]. The generative adversarial networks [23] provides a new idea for image compression. The generative adversarial networks was proposed by Goodfellow *et al.* based on the relevant principles of game theory. The core of generative adversarial networks is to generative model and discriminative model. The role of generative model is to generate new samples, while the role of discriminative model is to verify the authenticity of samples and determine the quality of generative samples. The optimization of network structure refers to the strategy of ‘‘binary minimax game’’ [24]. During the training process, keep the parameters of one model unchanged, change the parameters of another model, and repeat them alternately. Finally, the data generated by the model is close to the original sample. The application of generative adversarial networks has expanded from the initial image generation to various fields of computer vision [25], such as image recognition and video prediction.

II. RELATED WORKS

A. IMAGE COMPRESSION BASED ON DEEP LEARNING

Image compression is an important research method in the field of digital image processing, which has a broad application in video transmission, image redundancy removal and image stitching. Jiang [26] proposed a compression framework based on CNNs, the two CNNS are seamlessly integrated into an end-to-end compression framework, and has accurate reconstruction of decoded images. Yoo *et al.* [27] proposed a two-step framework for reducing blocking artifacts in different regions based on increment of inter-block correlation, which classifies the coded image into flat regions and edge regions. Sun and Cham [28] put forward the solution with the maximum posterior criterion. The distortion caused by coding is modeled as spatially related gaussian noise, and the original image is modeled as a high-order markov random field based on expert frame field. Prakash *et al.* [29] proposed a new CNN architecture specifically for image compression, which generates a label map that highlights semantically annotated areas. Cavigelli's *et al.* [30] took inspiration from deep neural networks developed for semantic segmentation and proposed a neural network with hierarchical skip connections and a multi-scale loss function for compression artifact suppression. These papers provide some inspiration for the algorithm design in this paper.

Although these algorithms utilize the powerful computational features of deep learning and save a lot of bandwidth, they ignore the comparison with the original image and result in low fidelity of partially compressed images. On the contrary, the proposed framework makes use of the characteristics of the generative adversarial networks to compare the compressed image with the original image in order to improve the visual quality of the image.

The framework of image compression includes several modules: encoder, quantizer, inverse quantizer, decoder and entropy coding [31]. The function of the encoder is to convert the image into the compression feature, and the decoder is to recover the original image from the compression feature. The encoder and decoder can be designed by convolution, pooling, nonlinear and other modules. Taking a three-channel picture of 768*512 as the input of the encoder, after forward processing, the compression feature occupying 96*64*192 data units can be obtained. If the compressed data are floating point numbers, the quality of the restored image will be the highest. But a floating-point number takes up 32 bits, and the number of bits per pixel is much larger after compression. The technique of quantization is used to convert floating point numbers into integer or binary numbers. At the decoding end, inverse quantization technology can be used to restore the transformed feature data to a floating point number, which can reduce the influence of quantization on the accuracy of the neural network and improve the quality of the restored image.

B. THE STRUCTURE OF GENERATIVE ADVERSARIAL NETWORKS

Image compression based on generative adversarial networks is a novel image compression method. Generative adversarial networks is a method of modeling according to the characteristics of training samples. The model contains two networks. the generator obtains the distribution of data in the sample and continuously generates new samples closer to the real sample. The discriminator is usually a binary classifier, which determines whether the input content is real data or generative samples, and finds the difference between generative samples and real samples. As a training framework, adversarial network does not require a new neural network structure. Some mature network models, such as RNN [32] or CNN [33], can be selected for each component of the framework.

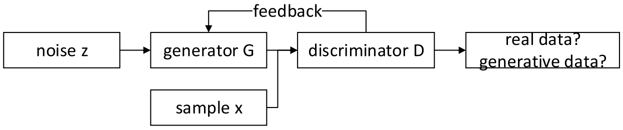


FIGURE 1. The model of generative adversarial networks.

The basic structure of generative adversarial networks is shown in Fig. 1.

In order to learn the generative distribution in the data set, the generator G takes the prior distribution of random noise P_Z as input to generate the sample distribution $P_G(z)$ approximate to the real data distribution $P_{data}(x)$, and try to make the performance $D(G(z))$ of the generative data $G(z)$ on the discriminator D consistent with the performance $D(x)$ of the real sample x, that is, the scatter metric of Jensen-Shannon [34] between x and $G(z)$ is the minimum. The divergence of Jensen-Shannon is used to measure the difference between two probability distributions. It is the deformation of KL divergence. The discriminator constantly extracts the characteristics of real samples, so that the output probability $D(G(z))$ tends to 0 and $D(x)$ tends to 1. Finally, the discriminator cannot distinguish whether the data belongs to real samples or generative samples. The mini-batch stochastic gradient descent training is used to generate the antagonistic network. The discriminator D is updated by the stochastic gradient ascending method, and the generator G is updated by the stochastic gradient descent method. The process of training is the process of game between the two, and the model eventually tends to the global optimum. The total loss function of G and D can be expressed as

$$L(D,G) = E_{x\sim P_{data}(x)}[logD(x)] + E_{x\sim P_{noise}(x)}[log(1-D(G(z)))] \rightarrow \min_G \max_D . \quad (1)$$

$G(z)$ is the image generated from using the input noise z, and x is the image in the real sample. E represents the expected value of the function. $x \sim P_{data}(x)$ means that x obeys the distribution of data. $x \sim P_{noise}(x)$ means that x obeys the distribution of noise variable.

GAN does not have fixed conditions, so on this basis a conditional generative adversarial networks [35] is designed(CGAN). In the modeling of generator G and discriminator D, CGAN uses conditional variables and additional information to add conditions to the model, which can supervise data generation and confrontation. Equivalent to converting unsupervised GAN into supervised CGAN, this condition variable y can be any type of data, such as category label, used to repair part of the data of the image. Every data in the CGAN model is associated with additional information s to a certain extent, in which the binary group (x, s) obeys the joint distribution $P_{x,s}$. The additional information s is integrated into the generator $G(z, s)$ and the discriminator $D(z, s)$ as part of the input layer. In the generative network, input noise z and additional information s constitute the joint hidden layer representation, and the structure is shown in Fig. 2. The total loss function of G and D can be expressed as

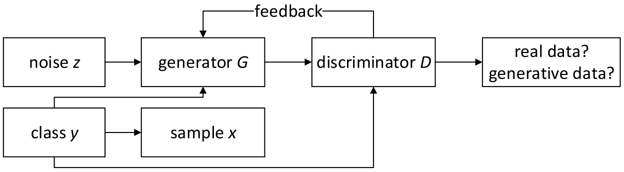


FIGURE 2. The model of conditional generative adversarial networks.

$$V(D,G) = E_{x\sim P_{data}(x)}[logD(x,s)] + E_{x\sim P_{noise}(x)}[log(1-D(G(z,s)))] \rightarrow \min_G \max_D . \quad (2)$$

s is additional information and i represents the uncompressed original image. E represents the expected value of the function. $x \sim P_{data}(x)$ means that x obeys the distribution of data. $x \sim P_{noise}(x)$ means that x obeys the distribution of noise variable.

C. SEMANTIC SEGMENTATION OF DISASTER IMAGES BY FULLY CONVOLUTIONAL NETWORKS

Fully convolutional networks (FCN) was first proposed in for solving the problem of semantic segmentation [36]. FCN mainly includes convolution layer, pooling layer and deconvolution layer, and its basic structure is shown in Fig. 3.

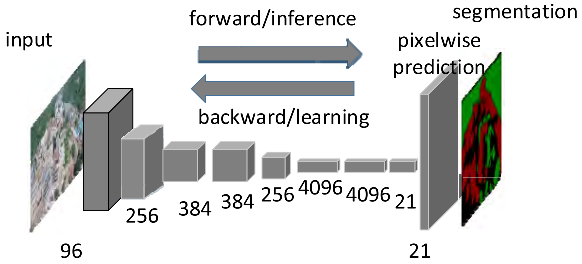


FIGURE 3. The basic structure of FCN.

The convolution operation [37] is used to extract features, and the image matrix X is convolved with a set of filters F. After convolution, the results as the input to nonlinear activation function $f(.)$, the formula is as follows:

$$Y_{i'j'} = f\left(\sum_{i=1}^{m}\sum_{j=1}^{n} F_{ij}X_{i+i',j+j'} + b\right). \quad (3)$$

$m \times n$ is the size of the convolution kernel, F_{ij} is the parameter of the convolution kernel, $X_{i+i',j+j'}$ are the input of the convolution layer, b is the bias, $Y_{i'j'}$ is the output of the convolution layer. $f(.)$ is the nonlinear activation function, and can be expressed as follows:

$$f(x_{ij}) = max\{0, x_{ij}\}. \quad (4)$$

The pooling layer reduces the dimension of image features and retains the most critical information. Max-pooling is used instead of mean-pooling because max-pooling retains more texture information, and mean-pooling retains more background information. It calculates the maximum value of the region within the range of $m \times n$ and the output $Y_{i'j'}$ of the pooling layer can be expressed as follows:

$$Y_{i'j'} = maxX_{i'+i,j'+j}, \quad i \in \{1, m\}, \; j \in \{1, n\}. \tag{5}$$

The deconvolution is the inverse operation of the convolution operation, which merely reverses the steps in the convolution transformation process. A prediction can be made for each pixel while the spatial information in the original input image is retained. The deconvolution is an up-sampling operation. The output size after the deconvolution operation is as follows:

$$O = (I - 1) \times s + p. \tag{6}$$

I is the input of deconvolution, s is the sliding stride of deconvolution kernel, p is the size of deconvolution kernel, and O is the output size.

The softmax classifier as the last layer in the network, which is used for the final classification and normalization. The formula of softmax function is as follows:

$$h_i = \frac{e^{X_i}}{\sum_{k=1}^{K} e^{X_k}}. \tag{7}$$

X_i represents the input sample of the output layer, K represents the total number of classes in the sample, and h_i represents the possibility that the classifier classifies the image as class i.

III. HIGH FIDELITY IMAGE COMPRESSION OF UAV BASED ON BASED ON GENERATIVE ADVERSARIAL NETWORKS IMPROVEMENT

A. THE BASIC WORKING PRINCIPLE OF COMPRESSION

The model of uav image compression used in complex disaster conditions is designed based on generative adversarial networks. It has the basic structure of image compression, including encoder E, decoder G, quantizer Q and inverse quantizer $\hat{Q}$. The encoder E converts the image to a compression form w with less bit storage space. The decoder G (also used as a generator) restores the information of the image by forming a reconstruction $G(Q(E(i)))$. The quantizer Q transmits the gradient back to the input data of the discriminator, and the inverse quantizer $\hat{Q}$ reconstructs the image based on the data provided by the decoder G. The effect of compression needs to consider both visual quality and compression rate. It can be expressed by the formula $\sqrt{\mu(E[L(i, G(z))])^2 + \delta H(\hat{w})^2}$, where L is a loss function to measure the visual similarity of the real image and the compressed image. In the formula, entropy $H(\hat{w})$ is the average information output of each gray level, and can represent the minimum bit rate w required for the gray level information in the image. The entropy can adjust the code rate of the compressed image through the boundary $log_2(Layer)dim(\hat{w})$. Weight δ and μ are used to adjust the compression effect of the model. The design of formula $\sqrt{\mu(E[L(i, G(z))])^2 + \delta H(\hat{w})^2}$ may cause that the visual quality is not optimal after image compressed, but the compression rate is also guaranteed as the evaluation of the compression effect. The value of the final formula tends to a certain range, and the image will achieve the better compression effect.

The decoder G generates the decoded image representation according to the vector z, and the discriminator finds the difference between the compressed image $\hat{i}$ and the real image i. Compression can be optimized by solving the saddle point of the function

$$\min_{E,G} \max_{D} E\left[f\left(D\left(i\right)\right)\right] + E\left[g\left(D\left(G\left(z\right)\right)\right)\right] + \sqrt{\mu(E[L(i, G(z))])^2 + \delta(H(\hat{w}))^2}. \tag{8}$$

E represents the expected value of the function. It is different from the Encoder E. i represents the uncompressed original image. z represents latent vector, and it is a combination of v and $\hat{w}$. v is the extracted noise. $\hat{w}$ is the quantized representation of the latent feature map w. G represents a generator and D represents a discriminator. L is a loss function for measuring the similarity between the original image i and the compressed image $\hat{i}$. Weight δ and μ are used to adjust the compression effect of the model. f and g are scalar functions. Here makes $f(y) = (y-1)^2$ and $g(y) = y^2$. $H(\hat{w})$ is the average information output of each gray level.

Since the last one term of formula (8) are not related to discriminator D, it will not affect the ability of discriminator to estimate samples. So we can rewrite equation (8) as follows:

$$\min_{E,G} \ell_{GAN} + \sqrt{\mu(E[L(i, G(z))])^2 + \delta(H(\hat{w}))^2}. \tag{9}$$

ℓ_{GAN} represents the minimum divergence on the generator G. The description of other variables has been explained in equation (8).

The vector z in the formula contains $\hat{w}$, which stores the information of disaster area image i. A key factor to improve the image compression quality is to set δ close to 0 and make the number of Layer or the dimension of $\hat{w}$ large enough, so that $\hat{w}$ can contain more code rate and the model can almost reconstruct the content of the disaster area image losslessly. At this time, the divergence between p_i and $P_{G(z)}$ tends to be zero, and the code rate loss caused by the adversarial network will not affect the overall quality of the disaster image.

The work of compression is to remove redundant data, but it also needs to ensure the fidelity of compressed images and produce compressed images more and more real [38]. For the damaged roads in the disaster image, encoder E cannot effectively obtain the data of this part, and generator G needs to use the extracted noise v and compression representation $\hat{w}$ to preserve the road texture as much as possible, rather than simply synthesizing the data of this part. If the road in the image looks like a fuzzy gray stripe, it will not provide accurate information to disaster relief workers. The generator G can use vectors to repair images. The variables in the formula are not fixed, but still have some association with the content of the original image. In this way, the underlying structure of the image remains consistent, that is, some information is shared between the input and output, which is conducive to improve the quality of the compressed image. At the end of the iteration, the discrimination accuracy on the data set is calculated. Once the discrimination accuracy is saturated, the training is stopped to prevent the model from overfitting.

The above design is not able to judge the regions of important and background in the image, and will not make additional processing for local data. Encoder E and generator G automatically handle the balance between the compression quality of the whole image and the removal of redundant data without any information guidance. However, the scene studied in this paper is relatively special. It is based on the complex disaster conditions, and has no high requirements for the visual quality of some regions in the image. It is hoped that the amount of data can be reduced as much as possible for post-processing, so that the amount of data in unimportant regions can be greatly reduced. Referring to the structural design of CGAN, this paper makes further improvement to the compression structure. The additional information s of the image i is semantic labels in a complex disaster scenario. The model needs to provide relevant information to the encoder E and the generative network G, and needs to combine semantics in the encoding or decoding of images. This design is called region of interest generative compression, which guides network confrontation according to semantic labels. Region of interest generative compression can specify important regions in disaster images, and establish compression quality standards for these regions during compression, or specify decompression requirements for certain regions during image reconstruction.

Providing semantic labels to the structure requires semantic segmentation of disaster images [39]. Different parts of the images are divided into different classes according to semantics [40], and the images provided are RGB three-channel images. The image is divided into nine classes, each with a corresponding RGB value. These 9 different RGB values are one-hot coding, and the output is a 9-channel encoded image. Each channel represents a specific class. Table 1 is the specific encoding scheme.

Before using generator to decode, bit allocation needs to be combined with heat map. Roads, houses and other areas are set as 1, while rivers, mountains, trees and other areas are set as 0, which is converted into a binary heatmap m with the same spatial dimension as $\hat{w}$. The region of 0 corresponds to the region that should be completely synthesized (i.e., background region), and the region of 1 corresponds to the region that should be retained (i.e., important region). The synthesized region still contains a small amount of data from the original image i. The original disaster image and semantic segmentation image in the data set are shown in Fig. 4.

TABLE 1. Image encoding scheme.

Class	Label	One-hot Coding
0	Road	[1 0 0 0 0 0 0 0 0]
1	Tree	[0 1 0 0 0 0 0 0 0]
2	Bare mountain	[0 0 1 0 0 0 0 0 0]
3	Building	[0 0 0 1 0 0 0 0 0]
4	Field	[0 0 0 0 1 0 0 0 0]
5	River	[0 0 0 0 0 1 0 0 0]
6	People	[0 0 0 0 0 0 1 0 0]
7	Fire or smog	[0 0 0 0 0 0 0 1 0]
8	Unclassified	[0 0 0 0 0 0 0 0 1]

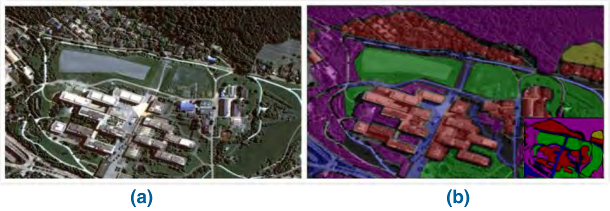


FIGURE 4. Original disaster image and semantic segmentation image. Fig (a) is original disaster image and Fig (b) is semantic segmentation image. Each region of the image has a different label. Some regions are labeled as region of interest, and some regions are labeled as region of background.

The semantic s and the image encoding are stored in different spaces, so the feature extractor extracts the data from it to the encoder. The data contains information to distinguish the region of interest and background, and then the generator G receives the data for further processing. The algorithm uses the semantic labels to guide the confrontation between networks. The compressed item $\hat{w}$ that should be synthesized in the background region is mostly set to 0, but a small part of data is also reserved for optimizing the visual effect. The algorithm uses the heat map to encode only the region of interest corresponding to item $\hat{w}$, which greatly reduces the bit rate required for image storage. The bit rate of the compressed item is much higher than that of the semantic label and heat map information, and the semantic label will not increase the data storage obviously. This method can save a lot of code rate of the disaster area image.

With the addition of semantic labels, there is a large difference in the amount of coding between the region of interest and the background region, which may lead to the lack of authenticity of the compressed image. We improve formula (9) and added the energy function of MRF to the loss function on the basis of GAN loss function. With the addition of the energy function, the pixels of the region of interest maintain continuity with the pixels of the region of background in image features at the maximum probability. The enhancement of image features between the region of interest and background makes accelerate the loss function, and further improves the training speed of the model. The energy function of MRF is defined as

$$K(x) = \beta \sum_{j,k} x_j y_k. \tag{10}$$

x_j is the pixel point of the region of interest, y_k is the pixel point of the region of background, and β is the weight parameter that makes the region of interest of the image consistent with the background region. The loss function of the final model is

$$\min_{E,G} \max_{D} E\left[K(i) + f(D(i))\right] + E\left[g(D(G(z)))\right] + \sqrt{\mu(E[L(i, G(z))])^2 + \delta(H(\hat{w}))^2}. \tag{11}$$

K is the energy function of MRF. f and g are scalar functions. L is a loss function for measuring the similarity between the original image i and the compressed image $\hat{i}$. The description of other variables has been explained in equation (8).

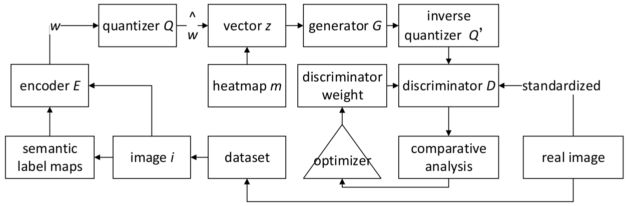


FIGURE 5. The work schematic diagram of region of interest generative adversarial networks.

In essence, the compression problem about the region of interest is still to remove redundant data from the image, and its working diagram is shown in Fig. 5. Firstly, the image is randomly selected from the data set and divided by semantic segmentation. Each part is marked and the corresponding features are extracted. The labels generated by the currently standardized image and the additional information after semantic segmentation are taken as the input of the encoder, and the output generates the encoding representation, which is then compressed by the quantizer. The compression representation is combined with the heat map to form a vector with conditional information. The data of the region of interest in the disaster image are reserved according to the vector, and other regions are synthesized as far as possible. The generative vector serves as the input of the generator, which generates the compressed image and gives it to the discriminator for judgment. The current standardized image and the compressed image generated by the generator are taken as the input of the discriminator, and the error result is used to make the judgment. Then it is fed back to the generator, and the weight of the generator and the discriminator is updated.

B. GENERATOR MODEL AND DISCRIMINATOR MODEL

The model of generator is essentially a decoder, which receives the vector generated by the encoder as input. It adopts the deconvolution neural network structure and consists of four convolution layers. The size of convolution kernel of each layer is 5 × 5, and the number of convolution kernel in the four layers is 512, 256, 128 and 64. Relu is adopted as the activation function in the structure.

The model of discriminator adopts convolutional neural network structure, including four convolutional layers and one fully connected layer. The convolution layer is used to compress the image, and then the full connection layer is used to classify the image to judge the authenticity of the image. The size of the convolution kernel of each convolution layer is 5 × 5, and the number of the convolution kernel is 64, 128, 256 and 512. Finally, a vector of 1024 is generated and processed by the sigmoid function to obtain a value in the range of 0 to 1, which is the probability of being identified as the original image. The model structure of generator and discriminator is shown in Fig. 6.

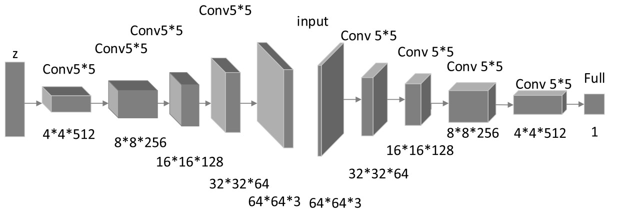


FIGURE 6. The model structures of generator and discriminator. The left side of Fig. 6. is the model structure of generator, and the right side is the model structure of discriminator.

IV. EXPERIMENTS AND DATA ANALYSIS

In order to verify the feasibility of the compression algorithm, the program is written based on python. GPU is used to enhance the training speed, and Tensorflow is used to realize deep learning algorithm. In the experiment, the disaster images collected by the uav over the years are used as the training set, and some images that do not meet the training conditions are removed. There are about 5,400 images in the training set. A large number of experimental data show that after the training set reaches 5400 images, adding more images can only bring insignificant parameter optimization, and some generative redundant parameters are not required for disaster scenarios. The test set also adopts the disaster area image collected by uav, but it is different from the training set. There are about 2500 images in the test set, and some images are shown in Fig. 7.

Adam gradient is selected to update method, and all samples in the training set 128 times are trained. The batch size of each processing in the training set is 1. The learning rate of generator and discriminator is 2e-4. In the network structure, the generator takes a 128 dimensional random sampling evenly distribution between 0 and 1 as input, and outputs a 64 × 64 compressed image; the discriminator takes a 64 × 64 compressed image and an original image as input, and the output is used to distinguish whether input image is compressed or original.

Firstly, the experiment needs to conduct semantic segmentation of the disaster image, analyze the key part of the information in the image, and use it as conditional information

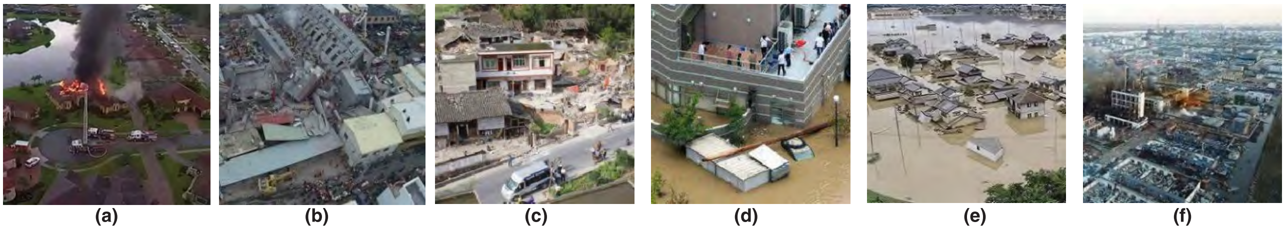


FIGURE 7. Part of the images in the test set. Fig (a) is fire image, and the image name in the test set is fire0054. Fig (b) and (c) are earthquake disaster images, and the image names in the test set are seismic0983 and seismic1106 respectively. Fig (d) and (e) are flood images, and the image names in the test set are flood1417 and flood1643 respectively. Fig (f) is explosion disaster image, and the image name in the test set is blast2412.

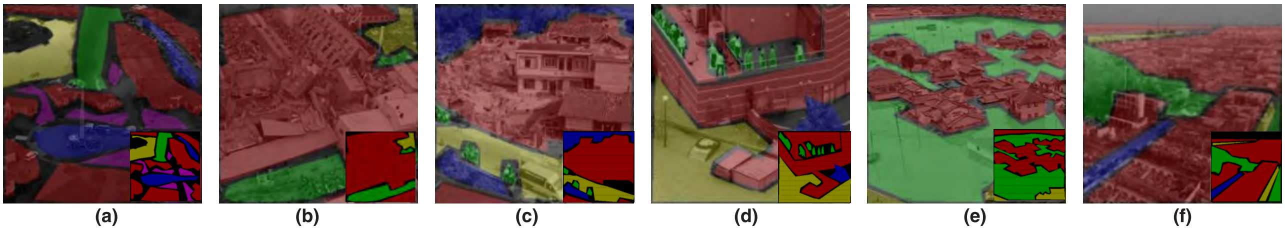


FIGURE 8. Semantic segmentation results of some disaster images. Different parts of the image are semantically divided into different classes. Fig (a) is the semantic segmentation of fire image. Fig (b) and (c) are the semantic segmentation of earthquake disaster image. Fig (d) and (e) are the semantic segmentation of the flood image. Fig (d) and (e) are the semantic segmentation of the explosion disaster image. These six images are semantic annotation images of Fig. 7.

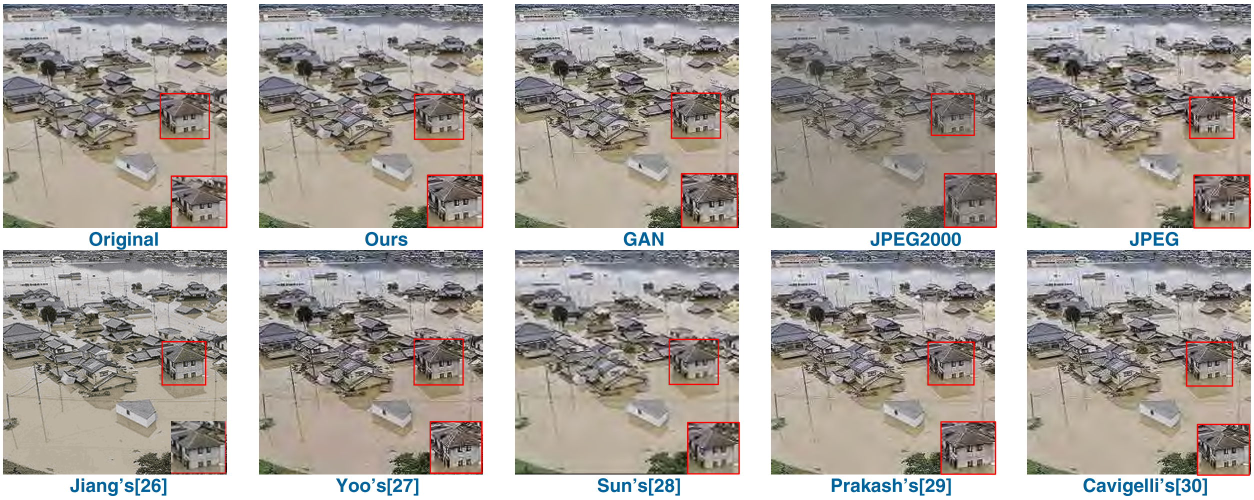
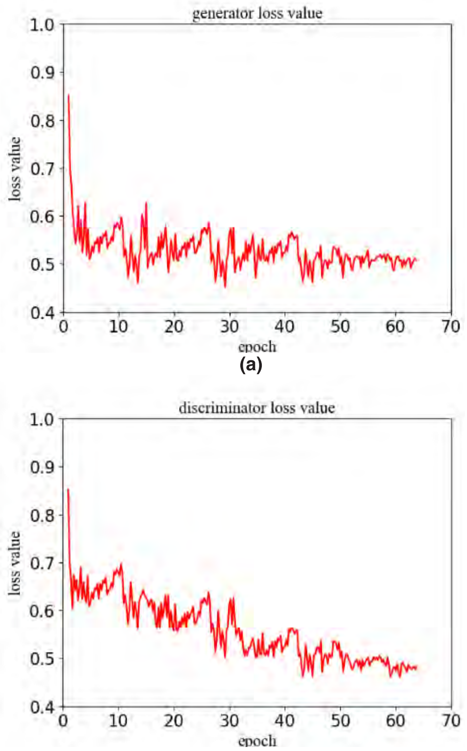


FIGURE 9. Comparison results of image compression for flood1643 in the case of BPP = 0.3. From left to right and top to bottom: Original image, the proposed algorithm (PSNR = 28.74 dB, MS-SSIM = 0.908), GAN (PSNR = 27.17 dB, MS-SSIM = 0.904), JPEG2000 (PSNR = 28.33 dB, MS-SSIM = 0.897), JPEG (PSNR = 25.14 dB, MS-SSIM = 0.827), Jiang's (PSNR = 28.45 dB, MS-SSIM = 0.899), Yoo's (PSNR = 26.27 dB, MS-SSIM = 0.862), Sun's (PSNR = 26.15 dB, MS-SSIM = 0.843), Prakash's (PSNR = 27.10 dB, MS-SSIM = 0.887) and Cavigelli's (PSNR = 26.59 dB, MS-SSIM = 0.876).

after processing. The semantic segmentation of some disaster images in the data set is shown in Fig.8. After continuous training, the discriminator cannot distinguish whether the image is original or compressed, the compression quality is constantly improved.

In order to further evaluate the effect of the proposed compression framework, the compression algorithm of this paper is compared with the standard compression methods (e.g., JPEG and JPEG2000). JPEG2000 is an image compression standard based on wavelet transform. Compared with JPEG, JPEG2000 has higher compression ratio and no block-blur defect. The proposed compression framework draws on some creative designs of Jiang's [26], Yoo's [27], Sun's [28], Prakash's [29] and Cavigelli 's [30], so the compression effects of these algorithms will also be used for comparison to prove that the design of the region of interest does enhance the compression effect of disaster images. In order to facilitate display, a disaster image is selected from the data set, and the compressed image of the improved algorithm is compared with that of other algorithms.



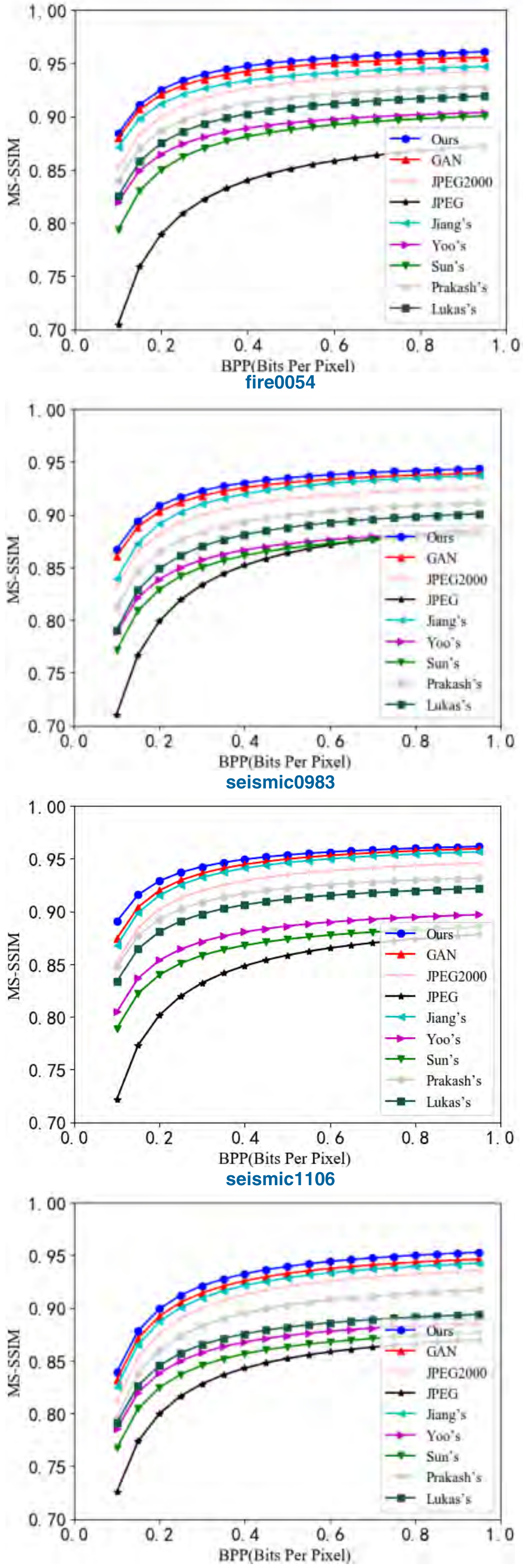
(a)

(b)

FIGURE 10. Training loss value of generators and discriminators under different epochs. Fig (a) is about the generator loss value. Fig (b) is about the discriminator loss value. Limited by the calculation conditions, the image is basically stable considering that the epoch is within 64. The value of epoch ranges from 0 to 64. The range of loss value is between 0.4 and 1.

The contrast effect is shown in Fig. 9. In image directly compressed by JPEG, significant blockiness occurs and the texture is blurred. Some artifacts appear when Yoo's [27] and Sun's [28] reconstruct the image. Prakash's [29], Cavigelli's [30] and JPEG2000 have better visual quality, but the edge of the area produce blurring effects. Jiang's [26] obtains better PSNR and SSIM, but the edge grayscale of the image changes drastically, and color saturation of the image is relatively low, which will lead to the compressed image looking uncoordinated. GAN has a higher visual quality, but cannot synthesize blocky or blurred spots. GAN does not highlight important areas such as houses. Compared to others compression algorithms, proposed compression algorithm retains more edge and texture details, and the resulting image has higher fidelity.

The traditional compression algorithm has obvious defects such as block effect in image reconstruction. The compression algorithm in this paper can retain the details of the road, house and other objects, and still be more similar to the original image after compression. Both the generator and discriminator use 64 epochs for training. The loss value decreases exponentially over 64 epochs, from 0.9 to about 0.5. Under different epochs, the loss value of the generator and discriminator of the algorithm is shown in Fig. 10.



fire0054

seismic0983

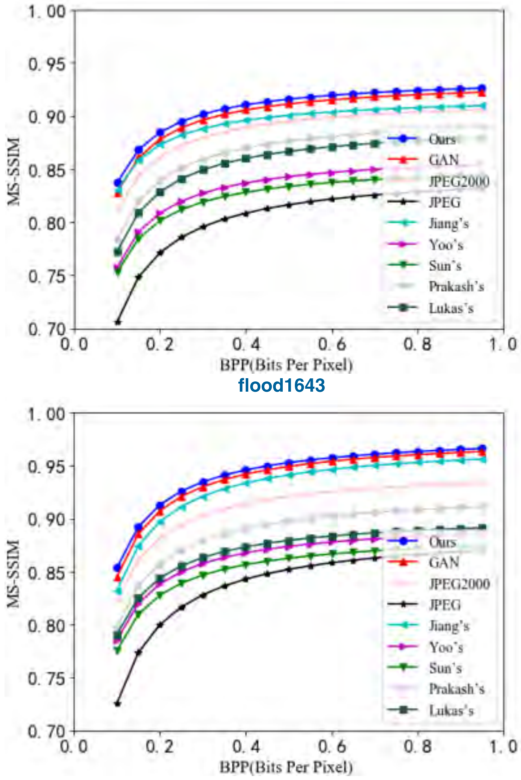
seismic1106

flood1417

FIGURE 11. Compressed quality evaluation results of nine algorithms. BPP is used to measure image resolution. The higher the BPP, the more colors are available. The measurement range of BPP is 0-1, and the value range of MS-SSIM is 0-1. The MS-SSIM of the all algorithms is basically stable when BBP is between 0.6 and 1.

TABLE 2. PSNR results of 9 contrast algorithms for BPP = 0.4 for multiple different disaster images.

Test images	fire0054	seismic0983	seismic1106	flood1417	flood1643	blast2412	average
Ours	26.42	30.09	31.07	31.14	30.56	29.04	29.72
GAN	25.47	29.09	29.05	28.96	28.89	27.17	28.11
JPEG2000	26.33	30.03	30.27	30.68	30.23	28.84	29.40
JPEG	22.58	26.45	27.27	27.53	27.43	25.64	26.15
Jiang's[26]	26.39	30.02	30.91	30.66	30.47	28.95	29.57
Yoo's[27]	25.41	28.69	28.57	28.69	28.71	26.77	27.80
Sun's[28]	24.78	27.96	28.17	28.15	28.06	25.83	27.16
Prakash's[29]	25.99	29.73	29.75	30.18	29.96	27.60	28.86
Cavigelli's[30]	25.55	29.12	29.52	29.63	29.94	27.49	28.49



flood1643

blast2412

FIGURE 11. ***(Continued.)*** **Compressed quality evaluation results of nine algorithms. BPP is used to measure image resolution. The higher the BPP, the more colors are available. The measurement range of BPP is 0-1, and the value range of MS-SSIM is 0-1. The MS-SSIM of the all algorithms is basically stable when BBP is between 0.6 and 1.**

The proposed compression algorithm preserves more image detail with a higher compression ratio, so that the image quality is not greatly damaged. We use MS-SSIM to further evaluate the performance of the algorithm. MS-SSIM is used to evaluate the visual quality of image compression algorithm. The higher the value is, the better the image reconstruction [41] will be. The comparison results are shown in Fig. 11.

From the six graphs in Fig. 11, it can be seen that when the BPP reaches about 0.9, the proposed compression algorithm, GAN, Jiang's [26], and JPEG2000 remain around 0.95, while Yoo's [27], Sun's [28], Prakash's [29], and Cavigelli's [30] remain around 0.90. JPEG remains around 0.85. Under the same BPP, the MS-SSIM of the proposed compression algorithm is slightly better than other algorithms. The reconstruction of the proposed algorithm and GAN algorithm is based on the distribution of data samples, and fidelity is more outstanding.

The values of PSNR and MS-SSIM for all compression methods are obtained by running the original author's source code and adjusting the parameters to the optimal value. Table 2 shows PSNR values of 6 disaster images in the test set compressed by various algorithms under BPP = 0.4. Table 3 shows MS-SSIM values of 6 disaster images in the test set compressed by various algorithms under BPP = 0.4. It can be seen that the proposed algorithm is the highest in terms of PSNR, and the average value reaches 29.72. The average value of JPEG2000 and Jiang's [26] also exceeds 29. In terms of the MS-SSIM, the proposed algorithm and GAN algorithm are superior to other algorithms.

Finally, the four image datasets CIFAR-10, HKU-IS, ECSSD and Cityscapes are used to evaluate the performance of the algorithm, and the effectiveness of the proposed algorithm is further verified. The classes used to provide semantic labels in these four data sets are different from the disaster image datasets, and the experiment has additionally dealt with them. Considering the two indexes of PSNR and MS-SSIM, it can be seen from Table 4 and 5 that compared with other algorithms, the PSNR and MS-SSIM values of the proposed algorithm remain the largest. It shows that the proposed algorithm has certain universality.

In this paper, the algorithm can seamlessly merge the content of regions of interest and regions of synthesis when compressed. In addition, compared with the same network structure without synthesis, the proposed network structure greatly reduces BPP. When the object has repetitive structure, the visual quality is basically not damaged, which can

TABLE 3. MS-SSIM results of 9 contrast algorithms for BPP = 0.4 for multiple different disaster images.

Test images	fire0054	seismic0983	seismic1106	flood1417	flood1643	blast2412	average
Ours	0.939	0.919	0.956	0.951	0.955	0.931	0.942
GAN	0.935	0.916	0.951	0.947	0.946	0.928	0.937
JPEG2000	0.924	0.891	0.945	0.937	0.924	0.919	0.927
JPEG	0.831	0.796	0.839	0.842	0.843	0.833	0.831
Jiang's[26]	0.926	0.893	0.949	0.941	0.945	0.927	0.930
Yoo's[27]	0.826	0.820	0.877	0.893	0.873	0.848	0.856
Sun's[28]	0.833	0.799	0.853	0.882	0.851	0.841	0.843
Prakash's[29]	0.878	0.872	0.901	0.912	0.909	0.869	0.880
Cavigelli's[30]	0.843	0.868	0.883	0.897	0.884	0.856	0.864

TABLE 4. Average PSNR results of JPEG2000, GAN and ours for BPP = 0.2,0.3,0.4 for CIFAR-10, HKU-IS, ECSSD and cityscapes.

Rate(bpp)	Bpp=0.2			Bpp=0.3			Bpp=0.4		
DataSets	JPEG2000	GAN	Ours	JPEG2000	GAN	Ours	JPEG2000	GAN	Ours
CIFAR-10	27.42	26.77	28.31	29.07	28.53	30.22	30.53	29.63	32.07
HKU-IS	27.18	26.57	28.16	28.79	28.34	29.58	30.11	29.57	31.18
ECSSD	26.79	26.14	27.47	28.18	27.68	29.06	29.72	29.29	30.89
Cityscapes	28.23	27.86	28.78	29.16	29.02	30.43	30.78	29.97	32.43

TABLE 5. Average MS-SSIM results of JPEG2000, GAN and ours for BPP = 0.2,0.3,04 for CIFAR-10, HKU-IS, ECSSD and cityscapes.

Rate(bpp)	Bpp=0.2			Bpp=0.3			Bpp=0.4		
DataSets	JPEG2000	GAN	Ours	JPEG2000	GAN	Ours	JPEG2000	GAN	Ours
CIFAR-10	0.836	0.847	0.867	0.875	0.891	0.901	0.904	0.929	0.943
HKU-IS	0.835	0.842	0.854	0.871	0.882	0.896	0.902	0.917	0.939
ECSSD	0.823	0.836	0.849	0.856	0.865	0.881	0.889	0.904	0.916
Cityscapes	0.842	0.855	0.868	0.877	0.906	0.912	0.913	0.932	0.957

generate images with high fidelity and save a lot of code rate. At the same code rate, the visual quality of the proposed algorithm is even better than some mature algorithms.

V. CONCLUSION

Firstly, this paper introduces the content of generative adversarial networks algorithm and analyses how the algorithm can be applied to image compression under complex disaster conditions. The generator generates the decompressed image, and the discriminator discriminates the difference between the compressed image and the real image. They compete with each other to produce images with better compression effect. By semantic labels, additional information is added to the model, which can preserve the data in important areas and compress the redundant data in non-critical areas on a large scale. Experimental results show that under the premise of ensuring image quality, the algorithm in this paper can indeed reduce the redundancy of data, and is significantly superior to common compression algorithms in terms of visual quality and image compression performance. It has a high fidelity and can provide convenience for later target detection. All regions of interest in disaster images are compressed in a similar way, but each region in the actual scene has different visual quality requirements. Therefore, it is the main content of the next step to consider setting different priorities for each region in the image and increasing the compression weight.

ACKNOWLEDGMENT

The authors would like to appreciate all anonymous reviewers for their insightful comments and constructive suggestions to polish this paper in high quality.

REFERENCES

[1] Y. Cui, S. Li, L. Wang, M. Sha, and Y. Shu, "Disaster event management based on integrated disaster reduction and rapid service platform," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 649–652.

[2] D. B. Bil and E. Konukseven, "Fully autonomous mini/micro scale UAV field experiences and image processing applications," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, May 2018, pp. 1–4.

[3] Z. Zhao, S. Wang, C. Jia, X. Zhang, S. Ma, and J. Yang, "Light field image compression based on deep learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[4] S. Kim and N. I. Cho, "Hierarchical prediction and context adaptive coding for lossless color image compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 445–449, Jan. 2014.

[5] I. Schiopu and A. Munteanu, "Residual-error prediction based on deep learning for lossless image compression," *Electron. Lett.*, vol. 54, no. 17, pp. 1032–1034, Aug. 2018.

[6] S. A. Alshehri, "Neural network technique for image compression," *IET Image Process.*, vol. 10, no. 3, pp. 222–226, 2016.

[7] Y. Zhang, C. Wu, J. Li, and M. Guo, "PCM: A parity-check matrix based approach to improve decoding performance of XOR-based erasure codes," in *Proc. IEEE 34th Symp. Reliable Distrib. Syst. (SRDS)*, Oct. 2015, pp. 182–191.

[8] D. Micciancio, "Locally dense codes," in *Proc. IEEE 29th Conf. Comput. Complex. (CCC)*, Jun. 2014, pp. 90–97.

[9] J.-J. Ding, I.-H. Wang, and H.-Y. Chen, "Improved efficiency on adaptive arithmetic coding for data compression using range-adjusting scheme, increasingly adjusting Step, and mutual-learning scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3412–3423, Dec. 2018.

[10] S. Kanda, K. Morita, and M. Fuketa, "Practical string dictionary compression using string dictionary encoding," in *Proc. Int. Conf. Big Data Innov. Appl.*, Aug. 2017, pp. 1–8.

[11] L. Ye and Z. Hou, "Memory efficient multilevel discrete wavelet transform schemes for JPEG2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 11, pp. 1773–1785, Nov. 2015.

[12] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[13] F. Wu, Z. Wang, Z. Zhang, Y. Yang, J. Luo, W. Zhu, and Y. Zhuang, "Weakly semi-supervised deep learning for multi-label image annotation," *IEEE Trans. Big Data*, vol. 1, no. 3, pp. 109–122, Sep. 2015.

[14] C. Wu, C. Luo, N. Xiong, W. Zhang, and T. Kim, "A greedy deep learning method for medical disease analysis," *IEEE Access*, vol. 6, pp. 20021–20030, 2018.

[15] P. P. Brahma, D. Wu, and Y. She, "Why deep learning works: A manifold disentanglement perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 1997–2008, Oct. 2016.

[16] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[17] J. Chen, Q. Jiang, Y. Wang, and J. Tang, "Study of data analysis model based on big data technology," in *Proc. IEEE Int. Conf. Big Data Anal. (ICBDA)*, Mar. 2016, pp. 1–6.

[18] B. Li, W. Xiong, O. Wu, W. Hu, S. Maybank, and S. Yan, "Horror image recognition based on context-aware multi-instance learning," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5193–5205, Dec. 2015.

[19] C. Tapsai, "Information processing and retrieval from CSV file by natural language," in *Proc. IEEE 3rd Int. Conf. Commun. Inf. Syst.*, Dec. 2018, pp. 212–216.

[20] E. Ahanonu, M. Marcellin, and A. Bilgin, "Lossless image compression using reversible integer wavelet transforms and convolutional neural networks," in *Proc. Data Compress. Conf.*, Mar. 2018, p. 395.

[21] G. Shomron and U. Weiser, "Spatial correlation and value prediction in convolutional neural networks," *IEEE Comput. Archit. Lett.*, vol. 18, no. 1, pp. 10–13, Jun. 2019.

[22] S. Wang, C. Wu, L. Gao, and Y. Yao, "Research on consistency maintenance of the real-time image editing system based on bitmap," in *Proc. IEEE 18th Int. Conf. Comput. Supported Cooperat. Work Design*, May 2014, pp. 689–694.

[23] X. Chen, C. Xu, X. Yang, L. Song, and D. Tao, "Gated-GAN: Adversarial gated networks for multi-collection style transfer," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 546–560, Feb. 2019.

[24] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, "Recent advances of generative adversarial networks in computer vision," *IEEE Access*, vol. 7, pp. 14985–15006, 2019.

[25] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, "Unsupervised learning based on artificial neural network: A review," in *Proc. IEEE Int. Conf. Cyborg Bionic Syst. (CBS)*, Oct. 2018, pp. 322–327.

[26] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 3007–3018, Oct. 2017.

[27] S. B. Yoo, K. Choi, and J. B. Ra, "Post-processing for blocking artifact reduction based on inter-block correlation," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1536–1548, Oct. 2014.

[28] D. Sun and W.-K. Cham, "Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2743–2751, Nov. 2007.

[29] A. Prakash, N. Moran, S. Garber, A. Dilillo, and J. Storer, "Semantic perceptual image compression using deep convolution networks," in *Proc. Data Compress. Conf. (DCC)*, Apr. 2017, pp. 250–259.

[30] L. Cavigelli, P. Hager, and L. Benini, "CAS-CNN: A deep convolutional neural network for image compression artifact suppression," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 752–759.

[31] C. Jia, X. Zhang, J. Zhang, S. Wang, and S. Ma, "Deep convolutional network based image quality enhancement for low bit rate image compression," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2016, pp. 1–4.

[32] M. E. Valle, "Complex-valued recurrent correlation neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1600–1612, Sep. 2014.

[33] J. Y. Cheong and I. K. Park, "Deep CNN-based super-resolution using external and internal examples," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1252–1256, Aug. 2017.

[34] L. Fan, "Supervised binary hash code learning with Jensen Shannon divergence," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2616–2623.

[35] P. Hensman and K. Aizawa, "CGAN-based manga colorization using a single training image," in *Proc. 14th Int. Conf. Document Anal. Recognit.*, Nov. 2017, pp. 72–77.

[36] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[37] D. Zeng and M. Zhu, "Background subtraction using multiscale fully convolutional network," *IEEE Access*, vol. 6, pp. 16010–16021, 2018.

[38] C. Liu, J. Liu, D. Peng, and C. Wu, "A general multiobjective clustering approach based on multiple distance measures," *IEEE Access*, vol. 6, pp. 41706–41719, 2018.

[39] N. Zhao, C. Banerjee, and X. Liu, "Nano-scale context-sensitive semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3062–3066.

[40] C. Liu, G. Zhang, A. Zhou, and W. Chunxue, "Image segmentation framework based on multiple feature spaces," *IET Image Process.*, vol. 9, pp. 271–279, Nov. 2015.

[41] 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.

QIUHONG HU is currently a Postgraduate student of software engineering with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include neural networks and machine learning, especially in image denoising and image synthesis, in the field of computer vision.

CHUNXUE WU 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 a Professor with 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, and networked control systems.

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

NAIXUE XIONG received both Ph.D. degrees from Wuhan University (about sensor system engineering) and the Japan Advanced Institute of Science and Technology (about dependable sensor networks), respectively. He is currently an Associate Professor (3rd year) with the Department of Mathematics and Computer Science, Northeastern State University, OK, USA. Before he attended Northeastern State University, he was with Georgia State University, Wentworth Technology Institution, and Colorado Technical University about 10 years. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.

He published over 280 international journal papers and over 120 international conference papers. Some of his works were published in the IEEE JSAC, the IEEE or ACM transactions, ACM Sigcomm Workshop, the IEEE INFOCOM, ICDCS, and IPDPS. He has been a General Chair, Program Chair, Publicity Chair, PC Member, and OC Member of over 100 international conferences, and as a Reviewer of about 100 international journals, including the IEEE JSAC, the IEEE SMC (Part: A/B/C), the IEEE Transactions on Communications, the IEEE Transactions on Mobile Computing, and the IEEE Transactions on Parallel and Distributed Systems. He serves as an Editor-in-Chief, Associate Editor, or Editor Member for over 10 international journals (including Associate Editor for the IEEE Transactions on Systems, Man and Cybernetics: Systems, an Associate Editor for Information Science, the Editor-in-Chief of *Journal of Internet Technology* (JIT), and the Editor-in-Chief of *Journal of Parallel & Cloud Computing* (PCC), and a Guest Editor for over 10 international journals, including *Sensor Journal*, *WINET*, and *MONET*. He has received the Best Paper Award at the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08) and the Best Student Paper Award at the 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009).

Dr. Xiong is the Chair of ''Trusted Cloud Computing'' Task Force, IEEE Computational Intelligence Society (CIS), http://www.cs.gsu.edu/~cscnxx/index-TF.html, and the Industry System Applications Technical Committee, http://ieee-cis.org/technical/isatc/. He is a Senior Member of the IEEE Computer Society.