Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5344890, 14 pages
https://doi.org/10.1155/2021/5344890

Research Article

Human Detection via Image Denoising for 5G-Enabled Intelligent Applications

Hui Li,[1] **Hang Zhou**,[1] **Xiaoguo Liang**,[1] **Fen Cai**,[2] **Lingwei Xu**,[1,2,3] **Wei Kong**,[1] **and Ying Guo**[1]

[1]*School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266000, China*
[2]*Fujian Provincial Key Laboratory of Data Intensive Computing, Quanzhou, 362000 Fujian, China*
[3]*Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, 211189 Jiangsu, China*

Correspondence should be addressed to Lingwei Xu; xulw@qust.edu.cn

Received 9 September 2021; Revised 6 October 2021; Accepted 28 October 2021; Published 23 November 2021

Academic Editor: Junjuan Xia

5G technology strongly supports the development of various intelligent applications, such as intelligent video surveillance and autonomous driving. And the human detection technology in intelligent video surveillance has also ushered in new challenges. A number of video images will be compressed for efficient transmission; the resulting incomplete feature representation of images will drop the human detection performance. Therefore, in this work, we propose a new human detection method based on compressed denoising. We exploit the quality factor in the compressed image and incorporate the pixel_shuffle inverse transform based on FFDNet to effectively improve the performance of image compression denoising, then HRNet and HRFPN are used to extract and fuse high-resolution features of denoised images, respectively, to obtain high-quality feature representation, and finally, a cascaded object detector is used for classification and bounding box regression to further improve object detection performance. At last, the experimental results on PASCAL VOC show that the proposed method effectively removes the compression noise and further detects human objects with multiple scales and different postures. Compared with the state-of-the-art methods, our method achieved better detection performance and is, therefore, more suited for human detection tasks.

1. Introduction

With the emergence and development of 5G technology [1], it is widely used for data transmission, wireless communication, and other intelligent applications such as intelligent video surveillance and industrial Internet of Things [2–4]. Therefore, some researchers attempt to exploit deep learning technology to solve some problems in wireless communication [5] and industrial Internet of Things [6–8]. Besides, there are still many serious challenges in terms of human detection for intelligent video surveillance, a large number of images are processed by the JPEG compression algorithm for efficient transmission and storage, and the compression noise generated by this process causes image distortion, feature loss, and decrease in human detection performance. Although traditional denoising methods such as BM3D [9] and BM4D [10] and deep neural network denoising methods such as UDNet [11] and CBDNet [12] have achieved effective denoising to a certain extent, the denoising performance still needs to be improved. Besides, though the method based on deep learning for human detection can achieve more robust detection results than traditional methods like HOG [13] +SVM [14], and DPM [15], it still needs to further improve its detection accuracy for objects with small objects and variable scales [16, 17]. To solve the above problems, we propose a new human detection method based on image compression denoising. What is more, our method is evaluated and demonstrated on the PASCAL VOC benchmark datasets. Compared with the state-of-the-art methods, our method can achieve better detection performance after denoising and is, therefore, more suited for human detection tasks. The code will be released.

2. Related Work

Recently, in the field of denoising, the traditional methods based on manual design rules have limited capabilities; however, the deep learning technology based on the convolutional neural network (CNN) has made significant progress in the problem of image denoising [18]. For blind noise, DnCNN [19] is inspired by the idea of learning the residual of the mapping in the residual network instead of mapping itself to generate a noise map instead of the image after denoising, which can be widely applied to various noises. And for nonblind noise, FFDNet [20] adopts the noise level as a feature map and forwards it to the network together with the noise image after pixel_shuffle [21] inverse transformation, making full use of the noise level information. What is more, some scholars attempt to employ the Generative Adversarial Networks (GANs) proposed by Goodfellow and Pouget-Abadie [22] for the remarkable results in the field of image generation to denoise an image [23–26]. The core idea is to exploit MAE or MSE to retain the low-frequency part of the image, and then, it exploits GAN to restore the high frequency part of the image, but GAN is difficult to train and easily crashes during the training process. Yu et al. [27] attempt to incorporate reinforcement learning, resulting in a restoration algorithm RL-restore for a variety of degradation situations of universal images. Lehtinen et al. [28] investigated that for noise with expectation 0, it is possible to train a denoising network containing only dirty data noise to obtain clean images, using these images as an example.

In the object detection algorithm, the object detection method based on deep learning can obtain more robust detection results than traditional methods [29]. What is more, it has gradually formed a two-stage detection method based on the candidate region and a one-stage detection method based on regression. The two-stage object detection method based on the candidate region is mainly represented by the R-CNN series of algorithms. In R-CNN [30], selective search is first used to obtain the proposal region on the original image, then the original image is cropped according to the proposal region, then it is scaled to a fixed size to forward it into the convolutional neural network for extracting features, and finally, the support vector machine (SVM) is used for classification and bounding box prediction. Fast R-CNN [31] convolves the entire image to obtain a feature map and then performs other operations. Faster R-CNN [32] has changed the selective search to the region proposal network, which became an end-to-end training framework. Cascade R-CNN [33] is aimed at the problem that Faster RCNN only has single object detector, which leads to its poor detection performance. Therefore, the object detector is cascaded to improve the quality of prediction. On the other hand, although the one-stage detection method based on regression has lower accuracy than the two-stage detection method, it has a significant increase in speed. The one-stage detection method is represented by the YOLO [34, 35] and SSD [36] algorithms. It exploits only single CNN to directly predict the category and locate different objects in an end-to-end manner. However, if the object scale has changed, the detection results will have poor robustness. Therefore, Lin et al. propose Feature Pyramid Networks (FPN) [37], by fusing different levels of the feature map, so that a low-level feature map has strong semantic information and high-resolution characteristics to achieve high-quality detection of small object and multiscale object. In addition, HRNet [38] and HRFPN [39] improve on the common problem that an image's feature is first extracted and then restored to ensure feature extraction always maintains a high-resolution feature map.

Furthermore, Carion et al. [40] regard object detection as a direct set prediction problem and propose an end-to-end detection framework based on the encoder-decoder of transformer; Shen et al. [41] exploit residual structure and spatial sensitive entropy to reduce the negative impact of web images with noisy labels to a certain extent; Dai et al. [42] propose a dynamic detection head framework by unifying attention and object detection head; Wang et al. [43] propose a prediction-aware one-to-one label assignment to replace nonmaximum suppression postprocessing and achieve performance comparable to NMS.

However, each of the existing techniques in the literature has its limitations when used to include image denoising and human detection in complex scenes. Our approach here is to alleviate these difficulties. The contributions of our work are twofold.

Firstly, we establish a new denoising framework, which exploits the improved ResNet18 to extract the quality factor in the compressed image and then combines the pixel_shuffle inverse transform based on the FFDNet to effectively execute compression denoising.

Secondly, we introduce HRNet and HRFPN to extract and fuse high-resolution features of the denoised image, respectively, to obtain a high-quality feature representation of the object. And then, we exploit the cascaded object detector to classification and bounding box regression to make the inferred object box more accurate and further improve the object detection performance.

3. Method

Based on the characteristics of compressed noise and the existing problems of human detection, we deeply research the human detection algorithm with compressed noise images. In this paper, firstly, we obtain the denoised image based on deep learning technology; we establish a joint optimization solution for image noise removal and human detection by combining the extraction and fusion of high-quality features and gradually improving the human object estimation. It not only improves the subjective feeling of the image and the quantifiable performance index but also improves the performance of human detection (including the accuracy of classification and bounding box). A flowchart of the proposed framework is shown in Figure 1.

3.1. Human Image Compression Denoising. The image compression denoising method based on the convolutional neural network mainly consists of three steps, that is, noisy

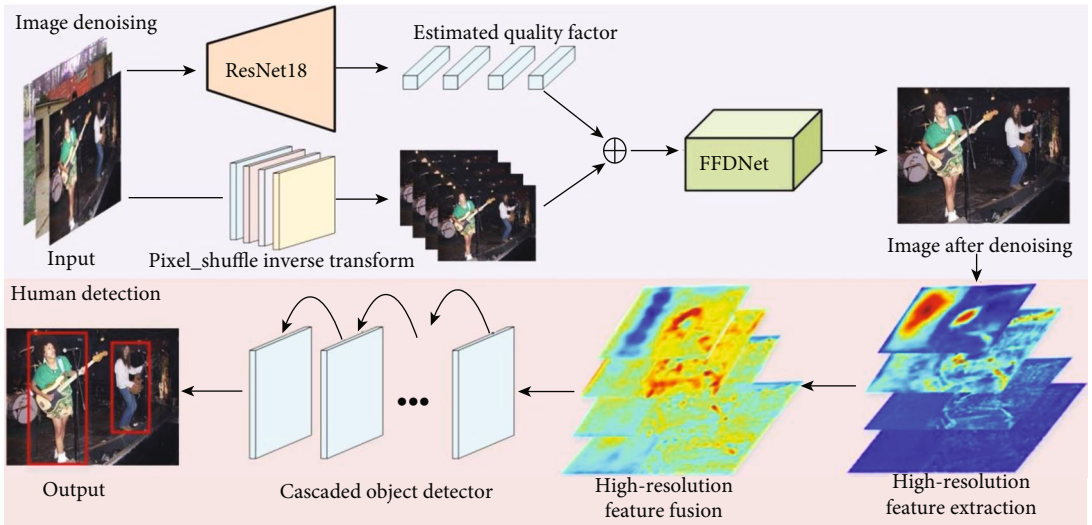


Figure 1: Flowchart of the proposed framework.

images are generated by lossy compression, quality factor estimation, and image denoising. Firstly, the JPEG compression algorithm is used to generate the training data with noise, then the value of the quality factor is estimated from the noisy image by the quality factor estimation network, and finally, original image containing noise and estimated quality factor values are used to form a set of feature maps as input to get the final image after denoising. And a flowchart of human image compression denoising is shown in Figure 2.

3.1.1. Image Lossy Compression. JPEG [44] (Joint Photographic Experts Group) compression algorithm is the most commonly used image compression algorithm on the Internet. It contains both lossy and lossless compression. What is more, lossy compression can be roughly divided into three processes: discrete cosine transform (DCT), quantization, and entropy coding. Discrete cosine transform is a process of transforming the image signal in the frequency domain to separate high-frequency and low-frequency information. The one-dimensional discrete cosine transform formula is as follows:

$$F(u) = c(u)\sum_{i=0}^{N-1} f(i)\cos\left[\frac{(i+0.5)\pi}{N}u\right],$$
$$c(u)\begin{cases}\sqrt{\frac{1}{N}}, & u=0,\\ \sqrt{\frac{2}{N}}, & u\neq 0,\end{cases} \tag{1}$$

where $f(i)$ is the vector to be converted, N is the vector length, and $c(u)$ is the coefficient of change. And the discrete cosine transform formula of a two-dimensional image is as follows:

$$F(u,v) = c(u)c(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s(i,j)\cos\left[\frac{(i+0.5)\pi}{N}u\right]\cos\left[\frac{(j+0.5)\pi}{N}v\right], \tag{2}$$

where $f(i,j)$ is the pixel value of point (i,j) and $c(u)$ and $c(v)$ are transformation coefficients. To speed up calculation, the restrictions are the same as formula (1). For the discrete cosine transform in an image, N is usually set to 80. Similarly, the calculation in formula (2) is usually completed in the form of a matrix to speed up the calculation:

$$A(i,j) = c(i)\cos\left[\frac{(j+0.5)\pi}{N}i\right], \tag{3}$$

where $c(i)$ is the inverse discrete cosine transform coefficient, and iterative traversal operation is turned into a parallel matrix operation through the calculated matrix $A(i,j)$. The corresponding inverse transformation formula is as follows:

$$f(i,j) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} c(u)c(v)F(u,v)\cos\left[\frac{(i+0.5)\pi}{N}u\right]\cos\left[\frac{(j+0.5)\pi}{N}v\right], \tag{4}$$

where $c(u)$ and $c(v)$ are the inverse discrete cosine transform coefficients and $F(u,v)$ is the frequency value of point (u,v). Since $A(i,j)$ is orthogonal, the matrix operation of the inverse discrete cosine transform is as follows:

$$f = A^{-1}F\left(A^{T}\right)^{-1} = A^{T}FA, \tag{5}$$

where A is the transformation matrix. Because inverse discrete cosine transform is lossless, therefore, it can restore

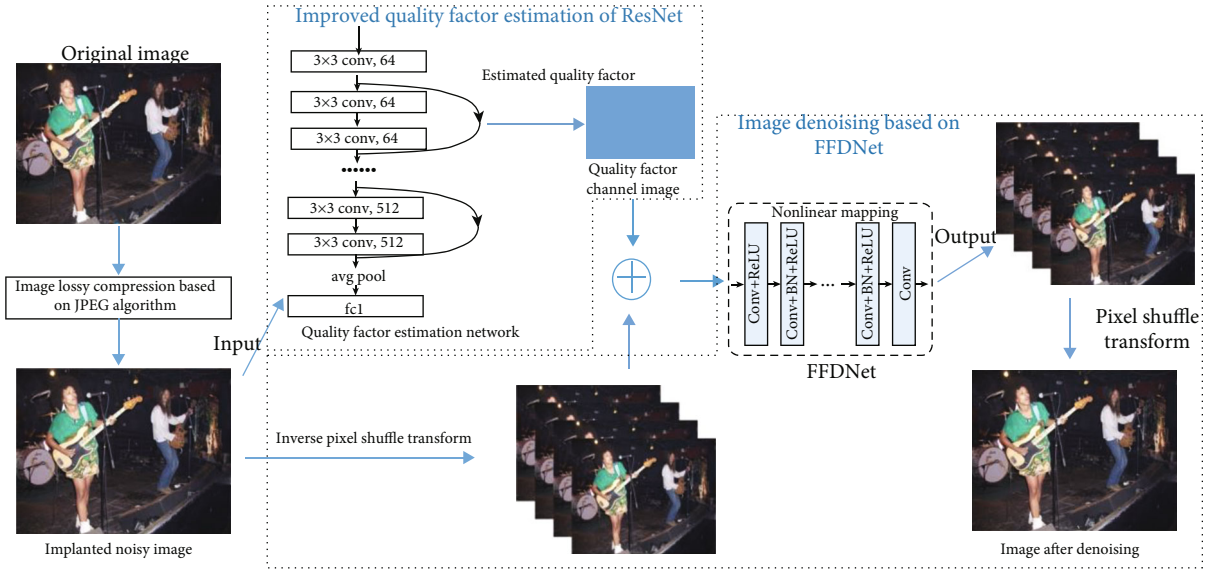


Figure 2: Flowchart of human image compression denoising.

the representation of the frequency domain to the pixel domain without loss.

Quantization is the only process that causes information loss in the JPEG algorithm. Since JPEG compresses the high-frequency part of image (i.e., image details) to reduce data, the image data are quantized and rounded to approximate values, which are different from the original image data, resulting in a permanent loss of information. Discrete cosine transform and quantization are both prepared for encoding. After the encoding is completed, the image compression is truly realized.

3.1.2. Quality Factor Estimation. In addition to standard R, G, and B channels, FFDNet can also receive noise channel N. Therefore, when the JPEG algorithm compresses an image to provide data for the noise channel, we exploit ResNet [45, 46] to learn the quality factor parameters; the quality factor is a key parameter that determines the degree of image distortion. However, estimating the quality factor from the noisy image is the task of regression; it is impossible to employ ResNet for classification directly. Therefore, we utilize the improved ResNet18 as the quality factor estimation network. The comparison diagram of the ResNet18 structure before and after the improvement is shown in Figure 3. The input is an image with compression noise, and the output is the estimated quality factor of each image. It contains a total of 17 convolutional layers and 1 fully connected layer. It uses a 3×3 convolution kernel and a convolution operation with a step size of 1. The number of convolution kernels for the first time is 64, and then, the step size becomes 3 at the 6, 10, and 14 layers; the number of convolution kernels is doubled. The last layer is a fully connected layer, which changes the number of neurons to 1, and there is an average pooling operation between the convolutional layer and the fully connected layer. The activation function of all activation layers in the improved ResNet18 is a ReLU function, and the improved function is shown in the following formula:

$$\mathrm{ReLU} = \begin{cases} x, & x > 0, \\ 0, & x \le 0. \end{cases} \tag{6}$$

(1) Loss Function. It is a regression problem to estimate the quality factor of the compressed image with an unknown quality factor during compression. Therefore, the most common square loss is used as the loss function. The square loss function is expressed as follows:

$$L(Y, f(X)) = \sum_{i=1}^{N} (Y - f(X))^2, \tag{7}$$

where X is the image with noise, f is the noise estimation network, $f(X)$ is the noise estimation value, and Y is the true value of the noise.

3.1.3. Image Denoising Network. Image denoising is mainly twofold: noise image preprocessing, and FFDNet denoising and generating an image postprocessing.

(1) Noise Image Preprocessing. To improve the denoising performance of the FFDNet, the noise image needs to be preprocessed. Firstly, the pixel_shuffle inverse conversion operation is executed, then the length and width of the original image P (C, W, H) is halved, and the number of channels becomes four times the original, i.e., the original image P is transformed into a tensor T1 with the shape ($4 * C$, $W/2$, $H/2$), where C is the number of original image channels (the number of RGB color image channels is 3, and the gray image is 1), W is the width of the image, and H is the height of the image. Then, the quality factor estimated by the improved ResNet18 is used as a channel, and the shape of the tensor T2 forming the channel is (1, $W/2$, $H/2$), and each value in it is a quality factor. Combining the

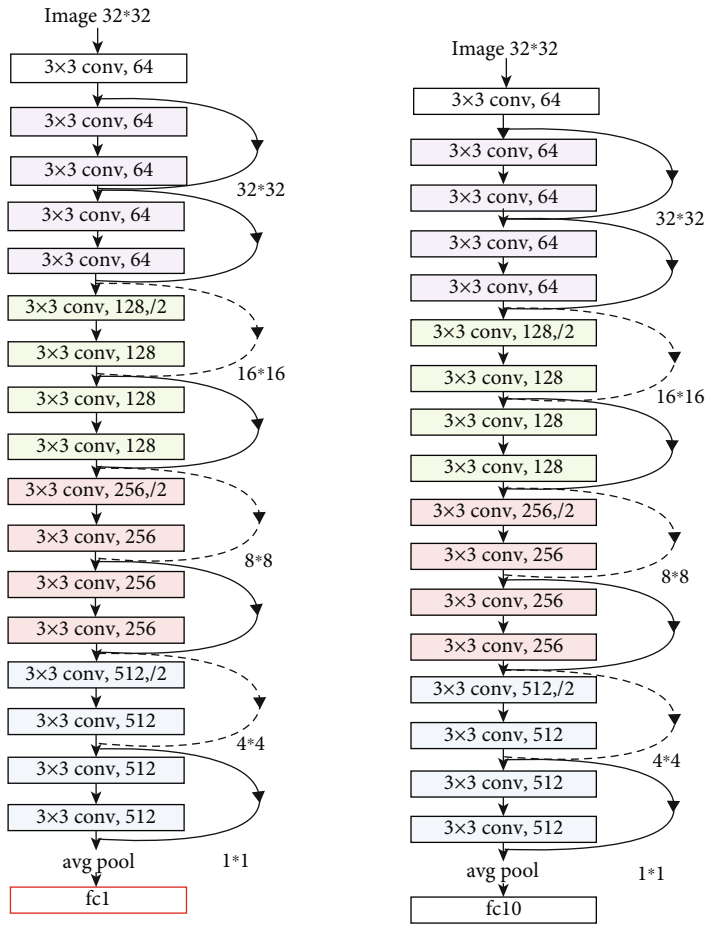


Figure 3: Comparison of ResNet18 network structure before and after improvement.

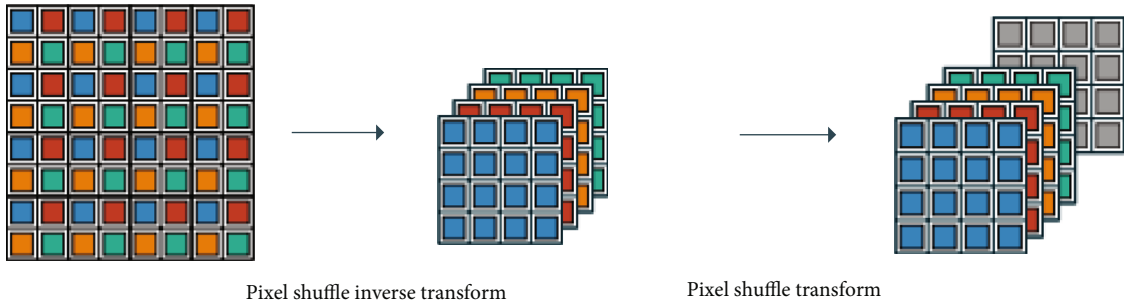


Figure 4: Flowchart of noise image preprocessing.

image pixel_shuffle inverse transformation, the $C * 4$ channel tensor and the tensor formed by the quality factor are concated, and the final tensor is $(C * 4 + 1, W/2, H/2)$. The schematic diagram of this process is shown in Figure 4.

(2) FFDNet Denoising and Image Postprocessing. The preprocessed noisy image is sent to the FFDNet. FFDNet is a fully convolutional network [47], which can generate clean images through end-to-end training. Its structure is shown in Figure 5. The network consists of $N + 2$ convolutional layers: the first convolutional layer uses ReLU as the activation function; each of the middle N layers has a convolutional layer, a BatchNormal layer, and a ReLU activation function. The last layer has only one convolutional layer without the BatchNormal layer and any activation function and directly outputs the denoised tensor. The direct result of the FFDNet output is a tensor of $(4C, W/2, H/2)$, not an image. In order to restore the tensor back to an image, a pixel_shuffle transformation is needed to restore $(C * 4, W/2, H/2)$ to (C, W, H), and this is the final denoised image.

(3) Loss Function. In the denoising stage of FFDNet, the absolute value loss is used as the loss function. The reason is that the square loss will blur the generated image for the

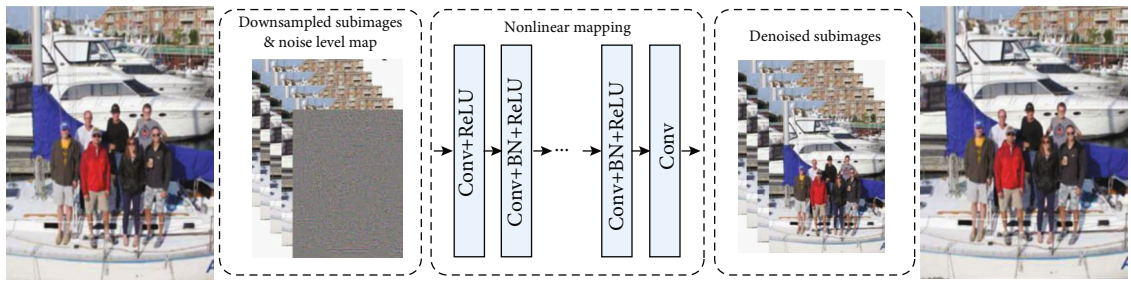


Figure 5: FFDNet structure diagram.

task of image denoising. The absolute value loss function is as follows:

$$L(Y, f(X)) = |Y - f(X)|, \tag{8}$$

where X is a tensor composed of a noisy RGB channel image and noise channel N, f is the denoising network, $f(X)$ is the image after denoising, and Y is the original image.

3.2. Human Detection. After image denoising, in order to further improve the performance of human detection, a human detection method based on high-resolution feature fusion is proposed in this paper. Firstly, HRNet is performed to extract high-resolution features, combined with HRFPN fusion multiscale features to obtain high-quality feature representation of the object, and then, the number of positive and negative anchors is balanced, and interest pooling is carried out in RPN to get the same size feature map. Finally, the cascaded object detector is used to execute multiobject detection. The technical route of this detection method is shown in Figure 6.

3.2.1. High-Resolution Feature Extraction and Fusion. In this paper, HRNet (High-Resolution Network) and HRFPN (High-Resolution Feature Pyramid Networks) are utilized to extract and fuse features in the image; these two operations yield a high-quality representation of features. Since HRNet can maintain a high-resolution representation of features during runtime, and the final feature maps contain both high-level semantic information and low-level cosmetic information about the object, so the final target features are more robust and rich. Its structure is shown in Figure 7. Firstly, it designs a path to obtain a high-resolution representation of the object; in this path, the resolution of the object feature map remains constant. And then, two parallel low-resolution paths are introduced in turn; meanwhile, the resolution of the feature map on the two parallel paths will remain constant; high-resolution features flow into the low-resolution layer after passing through several convolution layers, and finally, information is exchanged repeatedly between different resolution layers. In this way, the HRNet can extract richer information of the corresponding feature maps of human objects.

In addition, in order to improve the detection accuracy of a small object and variable object size, the HRFPN is performed to further integrate features obtained by HRNet; at last, we obtain a higher-quality feature representation of the image. Different from the Feature Pyramid Network (FPN) proposed specifically for multiscale object detection, HRFPN fuses all features from the beginning, and then, it scales layer by layer to fuse the information of each layer effectively so that higher quality features can be obtained. The comparison between FPN and HRFPN is shown in Figure 8. HRFPN first upsamples the feature maps of each layer with bilinear upsampling to a scale consistent with the largest feature map, stitches them together to form a new feature map, and then pools the feature maps in turn into feature maps of different scales. Since the proposed detection method is predicted at four scales, three pooling layers are also needed to obtain feature maps at different scales. And the pooling layer convolution kernel sizes are [2, 4, 8] in order; the pooling layer step size corresponds to the pooling layer convolution kernel size, which is also [2, 4, 8] in order.

3.2.2. RPN Extract ROIs. We are based on the Faster R-CNN framework, and then, we combine HRFPN based on backbone network HRNet with two stages of Faster R-CNN, and finally, multiscale RoIs and the multiscale object region are obtained according to multiscale RPN and the multiscale human object detector, respectively. The feature map obtained by HRFPN is fed into the RPN network. In each level, we employ the sliding to generate the region of interest. According to the region score and regression position, the multiscale region of interest (ROI) is obtained by clipping. The loss function at this stage is shown in the following formula:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls1}}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}} \sum_i p_i^* L_{\text{reg1}}(t_i, t_i^*), \tag{9}$$

where i represents the serial number of anchors in a small training batch, p_i is the probability of predicting the i-th anchor as the object, p_i^* is used to distinguish between positive and negative anchor points, t_i represents the bounding box predicted in the RPN stage, t_i^* represents the true bounding box position of the object, N_{cls} is the size of a small batch of training, N_{reg} is the number of anchors, and λ is the balance parameter and is used to balance classification loss and regression loss. The N_{cls1} classification loss function employs the exponential loss, and the N_{reg1} regression loss function exploits the smooth L_1 loss, as shown in the

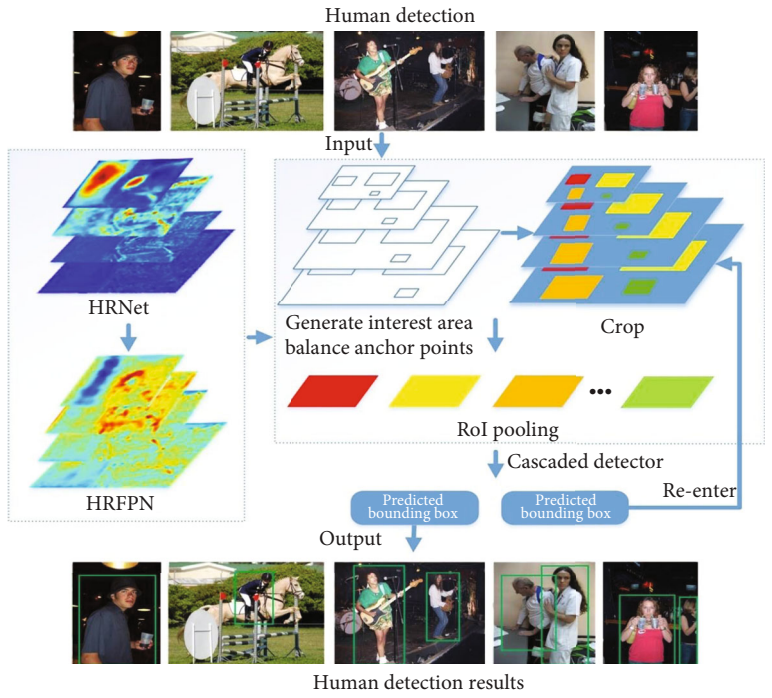


Figure 6: Technical flowchart of human detection method.

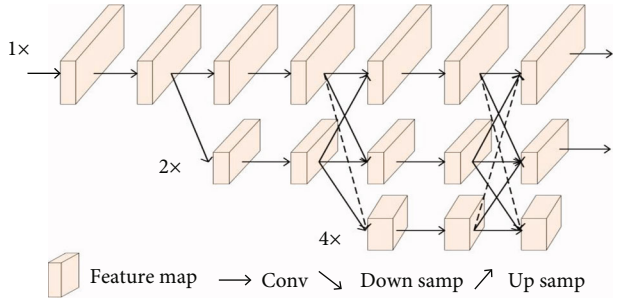


Figure 7: HRNet structure diagram.

following formula:

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1, \\ x| - 0.5, & \text{otherwise.} \end{cases} \tag{10}$$

In the formula, the parameter x is the difference between the predicted result and the true value.

In the RPN network, we divide the anchor regions into positive anchors and negative anchors according to the Intersection over Union (IoU) of the anchor region and the object ground truth. When the input image contains small objects, the number of negative anchors is far more than that of positive anchors; it will cause the problem of extracting too much background semantic information and ignoring the foreground object information; we adopt the method of randomly removing some negative anchors so that their number does not exceed three times that of positive anchors; the ratio of positive and negative anchors generated in the RPN stage is balanced, which can effectively improve the performance of the object detection network.

3.2.3. Cascaded Human Detector. In the human detection stage, the FPN is also utilized to extract human object features. We employ regions of interest from RPN, and we then matched them to their respective levels of the Feature Pyramid Network based on their size (length and width), and finally, we get the features of the object. Because the FPN stage has extracted enough deep feature maps, we only exploit ROI pooling to extract fixed-size object feature maps, and the final feature maps are fed into the Fast R-CNN object detector. Before the object detector, two fully connected layers are set up to get the confidence and region of the object. The loss function at this stage is shown in the following formula:

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \delta[u \geq 1]L_{\text{loc}}(t^u, v), \tag{11}$$

where L is the total loss function of the second stage object detection; p is the confidence of the predicted object; u is the true object category; t^u represents the predicted bounding box corresponding to the u category; v represents the true bounding box corresponding to the u category at the location; L_{cls} is the classification loss, which is the log loss corresponding to the real category u; δ is the balance parameter; and L_{loc} is the smooth L1 regression loss function, which is the level indicator function. When $u \geq 1$, the level indicator function value is 1; otherwise, the value of the function is 0, so that the bounding box loss can be calculated only for the foreground object, and the regression loss of the background cannot be calculated.

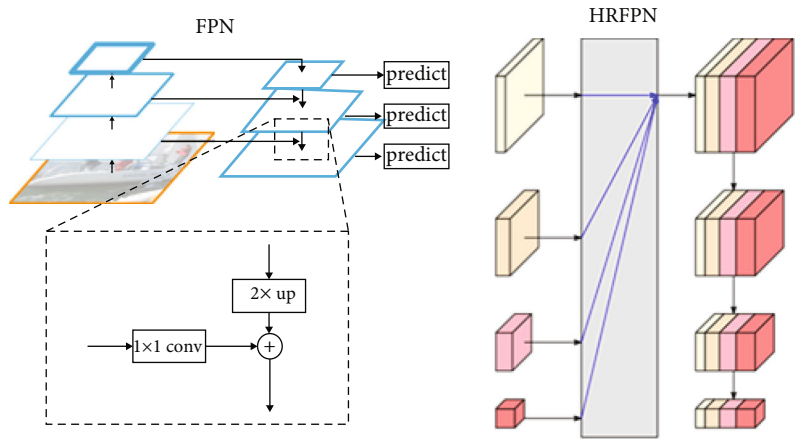


Figure 8: FPN and HRFPN process comparison diagram.

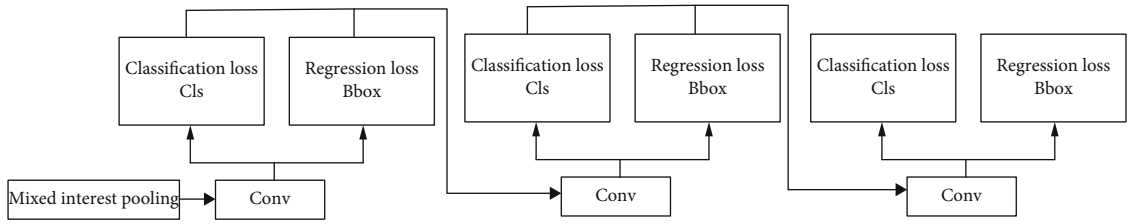


Figure 9: Cascaded human detector.

Table 1: Test results.

Iteration	50	100	150	200	250	300	350	400	450	…	1000
MSE	0.209	0.157	0.132	0.099	0.092	0.081	0.087	0.074	0.065	…	0.057

For the feature map after RoI pooling, classification and bounding box regression is performed on the cascaded object detector. Thus, the accuracy and robustness of object detection can be improved, and the categories of the predicted object and the positions of the corresponding bounding box can be obtained. The cascade operation is to exploit three object detectors with the same structure but different parameters, and it exploits the bounding box prediction results of the previous object detector as the new RPN proposal region, which is performed to crop the feature map. Finally, the cropped feature maps are fed into the next level of object detectors. The cascaded operation accumulates the loss of each object detector during training, where each object detector is composed as shown in Figure 9.

4. Experiment

4.1. Experiment Environment. We implement our network using the PyTorch framework. Furthermore, we utilize Ubuntu 16.04 and python 3.6 in our experiments, and the server CPU is Intel® Xeon® CPU E5-2678 V3 @ 2.50 GHz ∗ 2. Training is performed on NVIDIA GeForce GTX 1080 Ti GPU, which has 12 GB of memory. CUDA8.0 and CUDN6.0 were used to improve GPU utilization. Meanwhile, since we exploit two GPUs for training, we employ NCCL V2 to enable the GPU to communicate. At last, we use OpenCV for image processing.

Table 2: Test results of the network model.

Quality factor	PSNR (dB)		SSIM	
	Before denoising	After denoising	Before denoising	After denoising
35	30.89	32.71	0.910	0.928
45	31.70	33.54	0.921	0.938
65	33.23	35.20	0.933	0.947
75	34.40	36.20	0.946	0.958

4.2. Implementation Details. Our experimental dataset is the human body data from PASCAL VOC [48] with more than 6,000 pieces, divided by the original training and test sets of VOC; new human body images are selected to be trained with the training set and then tested with the test set in each round. In the human image denoising stage, we use the ADAM optimizer, and the training epoch is 10, the learning rate of the first 7 epochs is 1e-3, the learning rate of 8 and 9 is 1e-4, and the last epoch is 1e-6; a mini batch size is 128. In the stage of human target detection, we utilize stochastic gradient descent optimization algorithm, a learning rate of

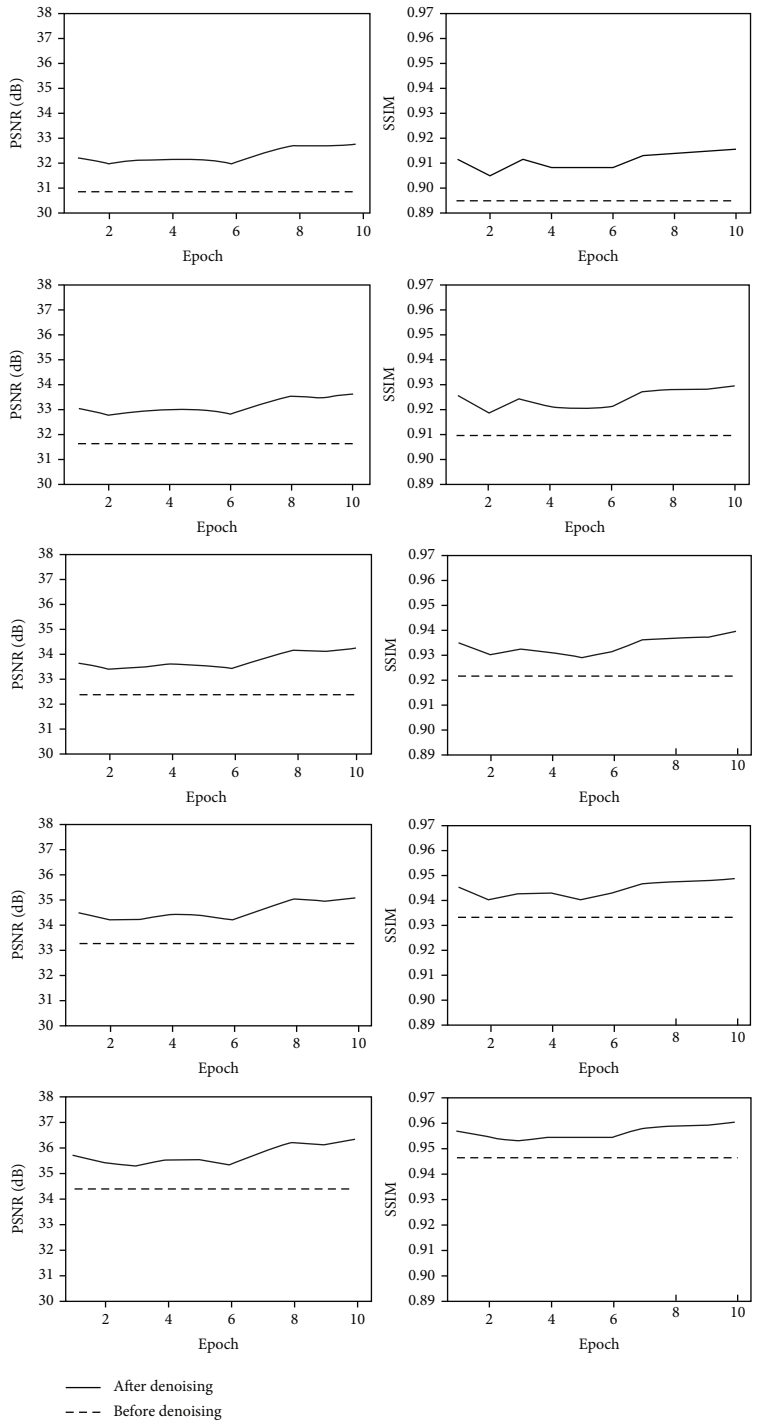


Figure 10: PSNR and SSIM value change curve before and after denoising.

0.02, the momentum is 0.9, the weight decay is 0.0001, and the training epoch is 75. In the above two stages, preprocessing operations are performed on the input images in order to increase the amount of data as well as to keep the training stable.

4.3. Evaluation Index. In our experimental, Peak Signal-to-Noise Ratio (PSNR) [49] and Structural Similarity (SSIM) [50] are used as judging metrics in the image compression denoising phase. PSNR is usually defined by mean square error (MSE). Firstly, given a pair of clean image I and noisy image K with the same size $m * n$, MSE is defined as follows:

$$\mathrm{MSE}(I, K) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(I_{i,j} - K_{i,j}\right)^2. \tag{12}$$

Then, PSNR is calculated as follows:

$$\mathrm{PSNR}(I,K)=10\cdot\log_{10}\left[\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}(I,K)}\right], \tag{13}$$

where MAX_I is the maximum pixel value of the image in the color space. Since MAX_K comes from the same color space, the maximum value of the two is equal, that is, $\mathrm{MAX}_I=\mathrm{MAX}_K$.

$$\begin{aligned} l(x,y)&=\frac{2\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1},\\ c(x,y)&=\frac{2\sigma_x\sigma_y+c_2}{\sigma_x^2+\sigma_y^2+c_2},\\ s(x,y)&=\frac{\sigma_{xy}+c_3}{\sigma_x\sigma_y+c_3}, \end{aligned} \tag{14}$$

where μ_x and μ_y are the mean values of x and y, respectively. The σ_x and σ_x are the variances of x and y, respectively. σ_x is the covariance of x and y.

The AP value is mainly used in the object detection phase to evaluate the detection algorithm proposed in this paper and also to measure the strengths and weaknesses of the algorithm by comparing it with other detection algorithms.

4.4. Results and Analysis

4.4.1. Denoising Results. To verify whether the improved ResNet18 can accurately estimate the value of the quality factor in the compressed image, Table 1 shows the mean square error between the value of the quality factor estimated using our method and its corresponding true value at different iterations; it can be seen that the mean square error between the estimated and true values of the compression quality factor is 0.099 at the number of iterations of 200, which is less than 0.1, then the improved ResNet18 network can be considered to be able to estimate the compression quality factor accurately.

In the training process based on FFDNet denoising, the L1 loss function (also known as the Mean Absolute Error) is used for training, and its formula is shown in 15 where x denotes a set of input values, y denotes the corresponding object value, n denotes the number of input values, x_i denotes the input value of the i^{th} sample, y_i denotes the object value of the i^{th} sample, and $f(x_i)$ denotes the estimate of the target value of the i^{th} sample.

$$\mathrm{MAE}(x,y)=\frac{1}{n}\sum_{i=1}^{n}|y_i-f(x_i)|. \tag{15}$$

Table 2 shows the results obtained by using the final trained network to test the image with compressed noise. It can be seen that for different quality factors between 35 and 75, the peak signal-to-noise is improved by about 2 dB after denoising, while the structural similarity is improved by about 0.15.

The before and after denoising results for images with compressed noise are shown Figure 10, where the dashed lines denote the PSNR and SSIM values of the images containing compression noise, the solid lines denote the PSNR and SSIM values after denoising, and the first to fifth rows of the figure show the PSNR and SSIM values before and after denoising obtained by the compression algorithm with quality factors of 35, 45, 55, 65, and 75, and it can be seen in the figure that both have improved with the increase in the number of training sessions, and it can be assumed that the performance of the network in removing the compressed noise is also increasing.

In Figure 11, the left column is the images with compressed noise, and then, the middle column represents the denoised images; at last, the right column represents the difference before and after image denoising. We observe the images of the left and right columns, it can be clearly seen that the distribution of noise is related to the content of the input, and its distribution is square with locality. Its change is obvious when the compression noise in the high frequency region of the image is removed. This is consistent with the JPEG compression process; it shows that the designed denoising network can indeed identify and remove JPEG compression noise.

Figure 11: Image comparison result display.

Table 3: Detection accuracy of human.

Detection method	AP
Baseline model	0.829
Baseline model+HRNet	0.849
Ours	0.861

Note: the results in the table are AP values, and the AP used is calculated according to the PASCAL VOC object detection standard.

Figure 12: Qualitative human detection results on PASCAL VOC.

4.4.2. Object Detection Results. To verify the object detection method proposed in this paper, Faster R-CNN+ResNet is used as the baseline model. Based on the baseline model, the backbone network is first replaced with HRNet, then the cascaded object detector is introduced, and the two are finally combined. Table 3 shows the accuracy results of human detection. The experiment first chooses Faster R-CNN+ResNet as the baseline model to determine a baseline for human detection, and its AP value is 0.829. Then, replacing the backbone network of the baseline model with HRNet, the AP value is 0.849, and the AP accuracy value is improved by 2% compared to the baseline model, indicating that HRNet can indeed extract high-resolution features to bring accuracy improvements. After that, when the backbone network uses ResNet, the single object detector is replaced with a cascaded object detector, and the AP value is 0.861. The accuracy of AP has been increased by about 1.2%, which is an increase of 3.2% compared to that of the baseline model. Therefore, it can be shown that the detection performance can be gradually improved by cascading object detectors and complement each other with HRNet to achieve high-performance human detection. Figure 12 shows part of the detection results of human objects. It can be seen that the detection method in this paper has better detection performance for human objects of different scales and poses in different complex scenarios.

To further illustrate that the performance of the human detection algorithm proposed in this paper is better than the current mainstream object detection algorithm, we exploit SSD, YOLO, Fast R-CNN, and other current mainstream object detection algorithms to conduct an experimental comparison with the algorithm in this paper. Various performance evaluation indicators are shown in Table 4. Comparative experiment results show that the AP value of the algorithm in this paper is significantly higher than that of other mainstream object detection algorithms, which also confirms the practicability of the algorithm in this paper.

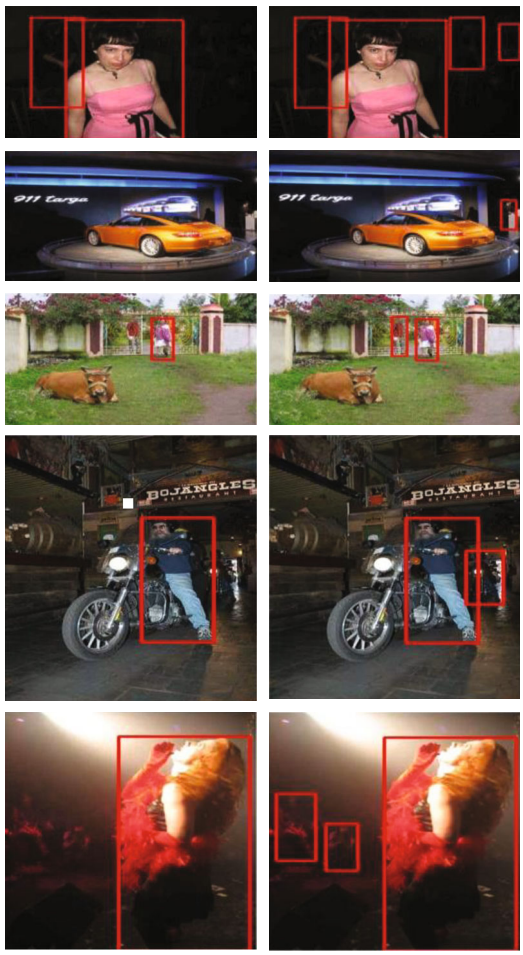
Table 4: Comparison of the results of mainstream object detection algorithms on the PASCAL VOC dataset.

Detection method	AP
SSD	0.635
YOLO	0.794
Fast RCNN	0.825
Faster RCNN	0.829
Mask RCNN	0.833
Cascade RCNN	0.843
Ours	0.861

Table 5: Comparison of detection results before and after denoising.

Detection method	Before denoising	After denoising
Baseline model	0.780	0.801
Baseline model+HRNET	0.795	0.807
Ours	0.803	0.832

4.4.3. Detection Results before and after Denoising. The above experiments are only carried out on the original PASCAL VOC test set. In order to show that the denoising algorithm can improve the performance of human detection, the quality factor of the JPEG compression algorithm is set to 40 and then applied to the test set, that is, the compression noise is implanted to imitate the distortion caused by the actual JPEG, thereby verifying the correctness of the algorithm. The experimental results are shown in Table 5. It can be seen



Detection results before denoising | Detection results after denoising

FIGURE 13: Comparison of visualization results before and after denoising.

from the table that the accuracy of human detection is about 2-3 percentage points higher than that before denoising, which shows that the information loss caused by the JPEG compression process not only produces visual artifacts but also interferes with the determination of target detection, and the feature distribution of the image after denoising is closer to that of the original image, thus improving the detection accuracy. The visualization results are shown in Figure 13.

5. Conclusion

In this paper, we present a human detection method based on image compression denoising. We exploit the improved ResNet to accurately estimate the quality factor in the image with compression noise and then combine the pixel_shuffle inverse transform based on FFDNet to effectively denoise; after that, we utilize cascaded object detectors for classification and bounding box regression based on high-quality feature representations obtained by HRNet and HRFPN. Finally, compared with the state-of-the-art methods, our method can achieve better detection performance after denoising and is, therefore, more suited for human object detection of intelligent video surveillance in 5G-enabled intelligent applications. In the future, we hope that the method proposed in this paper can make some contribution to the rapid development of 5G-enabled intelligent applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work is supported by the Opening Foundation of Fujian Provincial Key Laboratory of Data Intensive Computing under Grant BD202001 and Opening Foundation of Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education under Grant K93-9-2021-09.

References

[1] S. Tang, W. Zhou, L. Chen, L. Lai, J. Xia, and L. Fan, "Battery-constrained federated edge learning in UAV-enabled IoT for B5G/6G networks," *Physical Communication*, vol. 47, pp. 101381–101389, 2021.

[2] J. Xia, D. Deng, and D. Fan, "A note on implementation methodologies of deep learning-based signal detection for conventional MIMO transmitters," *IEEE Transactions on Broadcasting*, vol. 66, no. 3, pp. 744-745, 2020.

[3] K. He, Z. Wang, D. Li, F. Zhu, and L. Fan, "Ultra-reliable MU-MIMO detector based on deep learning for 5G/B5G-enabled IoT," *Physical Communication*, vol. 47, pp. 1–7, 2020.

[4] S. Tang, L. Chen, K. H. Xia, L. Fan, and A. Nallanathan, "Computational intelligence and deep learning for next-generation edge-enabled industrial IoT," *IEEE Transactions on Network Science and Engineering*, pp. 1–12, 2021, http://arxiv.org/abs/2110.14937.

[5] L. He, Z. Wang, T. Q. S. Quek, S. Chen, and L. Hanzo, "Deep learning-assisted terahertz QPSK detection relying on single-bit quantization," *IEEE Transactions on Communications*, pp. 1–12, 2021.

[6] Y. Guo, Z. Zhao, K. He, S. Lai, J. Xia, and L. Fan, "Efficient and flexible management for industrial Internet of Things: a federated learning approach," *Computer Networks*, vol. 192, article 108122, 2021.

[7] J. Xia, L. Fan, W. Xu et al., "Secure cache-aided multi-relay networks in the presence of multiple eavesdroppers," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7672–7685, 2019.

[8] H. Wang, L. Xu, Z. Yan, and T. A. Gulliver, "Low-complexity MIMO-FBMC sparse channel parameter estimation for industrial big data communications," *Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3422–3430, 2021.

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative

filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[10] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, 2013.

[11] S. Lefkimmiatis, "Universal denoising networks: a novel CNN architecture for image denoising," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3204–3213, Salt Lake City, UT, USA, 2018.

[12] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1712–1722, Long Beach, CA, USA, 2019.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, vol. 1, no. 12, pp. 886–893, 2005.

[14] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[15] D. Forsyth, "Object detection with discriminatively trained part-based models," *Computer*, vol. 47, no. 2, pp. 6-7, 2014.

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, Venice, Italy, 2017.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, 2009.

[19] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[20] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: toward a fast and flexible solution for CNN-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[21] W. Shi, J. Caballero, F. Huszar et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition.*, pp. 1874–1883, Las Vegas, NV, USA, 2016.

[22] I. Goodfellow and J. Pouget-Abadie, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 7, pp. 2672–2680, 2014.

[23] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Isgum, "Generative adversarial networks for noise reduction in low-dose CT," *Medical Imaging*, vol. 36, no. 12, pp. 2536–2545, 2017.

[24] N. Divakar and R. V. Babu, "Image denoising via CNNs: an adversarial approach," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1076–1083, Honolulu, HI, USA, 2017.

[25] Y. C. Li, N. F. Xiao, and W. L. Ouyang, "Improved boundary equilibrium generative adversarial networks," *IEEE access*, vol. 6, pp. 11342–11348, 2018.

[26] Q. Yang, P. Yan, and Y. Zhang, "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1348–1357, 2018.

[27] K. Yu, C. Dong, L. Lin, and C. C. Loy, "Crafting a toolchain for image restoration by deep reinforcement learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2443–2452, Salt Lake City, UT, USA, 2018.

[28] J. Lehtinen, J. Munkberg, J. Hasselgren et al., "Noise2Noise: learning image restoration without clean data," *International Conference on Machine Learning*, pp. 2965–2974, 2018.

[29] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: a survey," *Pattern Recognition*, vol. 51, pp. 148–175, 2016.

[30] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 23–28, Columbus, OH, USA, 2014.

[31] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, IEEE Press, Santiago, Chile, 2015.

[32] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, 2018.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[34] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, 2016.

[35] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, 2017.

[36] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Computer Vision – ECCV 2016*, pp. 21–37, Springer, Cham, 2016.

[37] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, Honolulu, HI, USA, 2017.

[38] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5693–5703, Long Beach, CA, USA, 2019.

[39] B. Cheng, B. Xiao, J. Wang, H. Shi, T. S. Huang, and L. Zhang, "HigherHRNet: scale-aware representation learning for bottom-up human pose estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5386–5395, Seattle, WA, USA, 2020.

[40] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *European Conference on Computer Vision*, vol. 12346, pp. 213–229, 2020.

[41] Y. Shen, R. Ji, Z. Chen et al., "Noise-aware fully Webly supervised object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11323–11332, Seattle, WA, USA, 2020.

[42] X. Dai, Y. Chen, B. Xiao et al., "Dynamic head: unifying object detection heads with attentions," in *Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021.

[43] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, and N. Zheng, "End-to-end object detection with fully convolutional network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021.

[44] G. Hudson, A. Léger, B. Niss, I. Sebestyén, and J. Vaaben, "JPEG-1 standard 25 years: past, present, and future reasons for a success," *Journal of Electronic Imaging*, vol. 27, no. 4, article 040901, 2018.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision – ECCV 2016*, pp. 630–645, Springer, Cham, 2016.

[47] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2015.

[48] H. Xian-Hua, C. Yen-Wei, and R. Xiang, "Multi-class object recognition by fusion of image descriptors: classification evaluation of PASCAL VOC challenge database," *Ice Technical Report*, vol. 109, pp. 103–108, 2009.

[49] Q. Huynh-Thu and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," *Telecommunication Systems*, vol. 49, no. 1, pp. 35–48, 2012.

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.