





# Deep Residual Learning-Based Enhanced JPEG Compression in the Internet of Things

Han Qiu , *Member, IEEE*, Qinkai Zheng , Gerard Memmi, *Member, IEEE*, Jialiang Lu ,  
Meikang Qiu , *Senior Member, IEEE*, and Bhavani Thuraisingham, *Fellow, IEEE*

**Abstract**—With the development of big data and network technology, there are more use cases, such as edge computing, that require more secure and efficient multimedia big data transmission. Data compression methods can help achieving many tasks like providing data integrity, protection, as well as efficient transmission. Classical multimedia big data compression relies on methods like the spatial-frequency transformation for compressing with loss. Recent approaches use deep learning to further explore the limit of the data compression methods in communication constrained use cases like the Internet of Things (IoT). In this article, we propose a novel method to significantly enhance the transformation-based compression standards like JPEG by transmitting much fewer data of one image at the sender's end. At the receiver's end, we propose a two-step method by combining the state-of-the-art signal processing based recovery method with a deep residual learning model to recover the original data. Therefore, in the IoT use cases, the sender like edge device can transmit only 60% data of the original JPEG image without any additional calculation steps but the image quality can still be recovered at the receiver's end like cloud servers with peak signal-to-noise ratio over 31 dB.

**Index Terms**—Discrete cosine transform, deep residual learning, image compression, Internet of Things (IoT), JPEG.

## I. INTRODUCTION

**D**URING the last decades, the development of big data has enabled many advanced data-oriented techniques such as artificial intelligence [1] and cognitive computing [2]. Such techniques, like cognitive computing, require transmission of a large amount of data including images or videos to build

cognitive models up. Nowadays, many such techniques have been deployed on practical use cases based on edge computing or the Internet of Things (IoT) [3]. Therefore, there is an increasing need for more efficient and secure data transmission for IoT use cases with communication or computing resource constraints.

The traditional efficient and secure data transmission relies on data compression methods performed at the sender's end and decompression methods at the receiver's end. Particularly, the compression methods for the multimedia content are lossy compression by selectively ignoring part of the unimportant part based on information theory [4]. For instance, discrete cosine transform (DCT) [5] is the most widely used transform in multimedia coding, which covers image and video coding standards such as JPEG, MPEG-1/2, MPEG-4, AVC/H.264, and the more recent high efficiency video coding (HEVC) [6]. DCT is a Fourier-like transform to compact most energy of a highly correlated discrete signal into a few coefficients. Deploying DCT in the multimedia data compression is basically to determine the importance levels of the DCT coefficients considering the visual effects and then to use the quantization step to compress these coefficients with unrecoverable loss.

One approach to improve the data compression before transmission is to ignore more data (e.g., high-frequency coefficients) at the sender's end and to rebuild the loss at the receiver's end. With such an approach, the amount of data to be transmitted can be reduced which is beneficial for IoT scenarios under the constraint of limited communication resources. For instance, in [7], the sensors collect images and send them to central servers, such an approach can help to reduce the amount of data during transmission.

One ground truth of the multimedia compression is that compressing more high-frequency coefficients will achieve a higher compression ratio but lead to worse image quality. JPEG also has different quantization tables that decide how many high-frequency coefficients will be kept according to compression requirements. However, on the other hand, compression of the low-frequency coefficients especially dc coefficients could also be a possible approach to achieve a higher compression ratio since dc coefficients count for more than 30% of the storage space in JPEG image. For instance, Chen *et al.* [8] showed a video compression method based on the prediction of the dc coefficients. Also, [9] showed dc coefficients of JPEG images can be guessed with all the rest DCT coefficients. These research approaches inspire that the JPEG image could be significantly further compressed by keeping only ac coefficients at

Manuscript received December 13, 2019; revised February 16, 2020 and April 28, 2020; accepted May 10, 2020. Date of publication May 14, 2020; date of current version November 20, 2020. Paper no. TII-19-5331. (Corresponding author: Meikang Qiu.)

Han Qiu, Qinkai Zheng, and Gerard Memmi are with the Department of Network and Computer Science, Telecom Paris, 91120 Paris, France (e-mail: han.qiu@telecom-paris.fr; qinkai.zheng@telecom-paris.fr; gerard.memmi@telecom-paris.fr).

Jialiang Lu is with the SPEIT, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jialiang.lu@sjtu.edu.cn).

Meikang Qiu is with the Texas A&M University Commerce, Commerce, TX 75428 USA (e-mail: qiumeikang@yahoo.com).

Bhavani Thuraisingham is with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: bhavani.thuraisingham@utdallas.edu).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TII.2020.2994743

the sender's end if the dc coefficients can be then recovered at the receiver's end. However, the two main approaches introduced in previous works (see Section II) for recovering dc coefficients are all suffering from the errors at the recovery steps [9], [10].

More recent approaches as shown in [11] are trying to improve the detailed quality of images with deep learning (DL) methods. Such work aimed to improve the image quality by rebuilding the detail information. In the real world IoT use cases, such a method can let the sender ignore more high-frequency coefficients, which can reduce the amount of data during transmission. However, the research on ignoring and rebuilding the low-frequency coefficients with DL methods is missing.

In this article, we propose a novel two-step method to achieve an accurate recovery for dc coefficients. First, we improved the state-of-the-art dc recovery method based on classic signal processing algorithms that can recover high-quality images with less visual artifacts. Then, inspired by the deep residual learning method, we deployed the DL model [12] to remove the visual artifacts that can allow the high-quality image recovery at the receiver's end.

Our main contributions include: first, we improve the state-of-the-art dc coefficient recovery method; second, we use the DL model to further optimize the dc coefficient recovery problem in the case that only AC coefficients are kept in a standard JPEG compression; third, with our method, an enhanced JPEG compression method is proposed for the IoT use cases by reducing 40% of the data to transmit without any additional cost on the IoT node ends. Sample code and data is available at <https://github.com/Stanislas0/Enhanced-JPEG-compression> to support further research.

The rest of this article is organized as follows. In Section II, research background is presented. In Section III-A, we illustrate how to use the proposed method. In Section III, we present the preprocess dc recovery method. In Section IV, we present the DL model. In Section V, we explain the experiment details and evaluate the results. We give discussion and future work in Section VI and conclusion in Section VII.

## II. RESEARCH BACKGROUND

In this section, first, we illustrate the initial research motivation by redefining the dc coefficients. Then, we list the main previous works for recovering the dc coefficients based on the block by block propagation methods and based on image spatial domain optimization, respectively.

### A. Practical Definition of the DC Coefficients

Most multimedia compression standards in frequency domain apply the DCT transform to small blocks sequentially to reduce the overall time complexity of the transformation. The JPEG standard [13] deploys DCT algorithms on 2-D (DCT-2-D) at a small block level of size  $8 \times 8$  pixels. This DCT-2-D for one  $8 \times 8$  block will generate 64 coefficients where each coefficient carries distinct information of the transformed signal. Among all DCT coefficients, the first one (i.e., the one with the lowest frequency) is called the dc coefficient while the rest are called ac coefficients. dc coefficient is considered as the special one

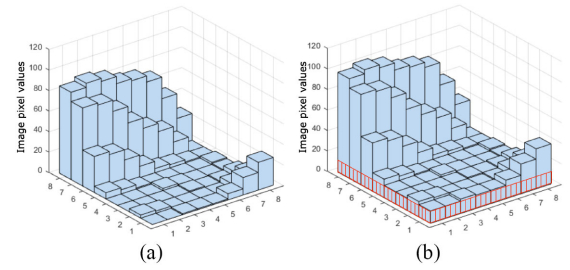


Fig. 1. Example of the definition of dc coefficient in a  $8 \times 8$  block: (a) Original distribution of pixel values. (b) Distribution of pixel values when dc coefficient increased: all pixels increased with the same value.

because it carries the average intensity of the transformed signal, which is the average value of the pixel values of the 64 pixels in this  $8 \times 8$  block. The compression algorithm will then compress the dc and ac coefficients with quantization steps to ignore many of the ac coefficients.

DCT has different types shown in [14], the most popular DCT algorithm is a 2-D symmetric variation of the transform that operates on  $8 \times 8$  blocks (DCT  $8 \times 8$ ) and its inverse (iDCT  $8 \times 8$ ). This DCT  $8 \times 8$  is utilized in JPEG compression routines [13] and has become an important standard in image and video compression steps.

According to the definition of the DCT transformation [13], the dc coefficient is the average value of the input elements. Thus, the dc coefficients of the DCT transform of image blocks represent the mean values of the pixel values in the corresponding image blocks. We illustrate this definition by an example of one  $8 \times 8$  block shown in Fig. 1. In the Fig. 1 (a), we list the pixel values on the z-axis for the original  $8 \times 8$  block. Then, we add the dc coefficient in the DCT result and apply the iDCT to get the pixel values with only the dc coefficient increased in Fig. 1 (b). The pixel value distribution is not changed in the Fig. 1 (a) and (b) but only every pixel value is added with the same value.

Thus, if all dc coefficients in all  $8 \times 8$  blocks change to zero, the relative difference of the pixel values remains. For the pixel value distribution on one JPEG image with all dc coefficients are zeros, every  $8 \times 8$  block still keeps the distribution inside the block but the average values of blocks are different. Based on this definition for the dc coefficients, the motivation of this research is to transmit only the ac coefficients in each block at the sender's end and to recover the corresponding dc coefficients at the receiver's end to realize higher compression ratio for JPEG images.

### B. Previous Work on DC Coefficient Recovery

The very initial question of the dc coefficients recovery was proposed by [10] since the early stage of selective encryption (SE) methods used to protect the dc coefficients of each  $8 \times 8$  block to further protect the image content. Such image protection methods are deployed on the JPEG images or other image formats with DCT transformation. The implementation of such methods showed a hard visual degradation for the images but the Uehara *et al.* [10] showed that the dc coefficients in an  $8 \times 8$  JPEG block can be guessed by the rest ac coefficients.

The guessing method is based on the observed statistical distribution of the image property in [10]: the difference signal

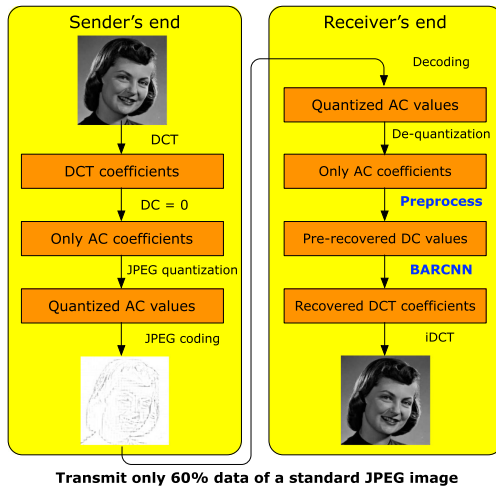


Fig. 2. Example of how the proposed method can be used with only 60% data transmitted for one JPEG image.

at the pixel level and the dc coefficients can be modeled as a zero-mean Laplacian distributed variate. The distribution is generally narrow with a small value of variance. Therefore, we could assume the value of variance as zero to determine the difference of the dc coefficients of the neighbor  $8 \times 8$  block. This method is experimented in [10] for the dc coefficients recovery in JPEG images and further improved in [9]. However, one issue in this method is that there will be tiny errors when we use the dc coefficient in one block to guess that in the neighbor block; moreover, these errors will further propagate block by block. As shown in [9], although the recovery results have an acceptable peak signal-to-noise ratio (PSNR) value and the image contents can be observed, the error propagation will lead to a lot of block artifacts in the image contents.

The other approach is based on the optimization algorithms [15] since this question can be solved by the linear programming (LP) technique for both the basic dc recovery problem and the general DCT coefficients recovery problem. Li *et al.* [15] also showed the recovery for the image content when more than 15 ac coefficients are also missing. In a bitmap case, even more than half of the low-frequency coefficients are missing, a rough recovery for the image content is still possible [16]. However, this approach does not suit our case since the image we want to deal with is JPEG images. In JPEG images, as there are the quantization and rounding step [17], most of the DCT coefficients are rounded to zeros (could be more than 90% of the DCT coefficients in the JPEG images as pointed in [9]). Although this LP approach can optimize the calculation and remove the edges due to the error propagation, the recovery results lost much detailed information. The summarized problem definition for this article and the main motivation of our research of using DL can be seen in Section III.

### III. ARCHITECTURE AND PROBLEM DESCRIPTION

In this section, we first illustrate how the designed method is working with an example shown in Fig. 2. Then, we point out the problems of error propagation existing in the traditional

approach based on the pixel value distribution theory in [10]. Along with this error propagation issue, the question to be solved in this article can be then summarized and described as a more precise statistical problem that could be solved by using DL models.

#### A. General Architecture

In this section, we show how to use the proposed method in the practical JPEG image transmission scenario. As shown in Fig. 2, at the sender's end, the JPEG image can be compressed without the dc coefficients. More specifically, for the standard JPEG compression process, we keep the same transformation step (DCT-2-D) and the encoding process. The only difference is that the dc coefficients in every  $8 \times 8$  will be removed and replaced with zeros. In this article, we keep only four dc coefficients of the  $8 \times 8$  blocks in the four corners of the image as the reference for the recovery step. For instance, for an image with a size of  $256 \times 256$ , there are  $32 \times 32$  blocks with each block contains one dc coefficient. For such an image, there are initially 1024 dc coefficients in total for the standard JPEG compression but now there suppose to be only four dc coefficients to be transmitted since 1020 dc coefficients are changed to zeros.

At the receiver's end, the JPEG decoding process is first performed to get the correct ac coefficients and the four dc coefficients of the four corner blocks. A preprocessing step will be deployed to have a prediction of the missing dc coefficients. This preprocess step proposed in this article is based on improving the previous work [9], which can roughly recover the dc coefficients by using the pixel value distribution theory in [10]. However, as pointed in [9], there will be block artifacts (not noise) in the recovered image due to the error propagation. Then, a deep residual learning model based on neural network, which is proposed in this article, will be used to further recover the dc coefficients precisely by removing the block artifacts and provide better image quality. Through this operation, the transmission data for a  $256 \times 256$  JPEG image are highly reduced without and additional cost on the sender's end like IoT device (60% compared with JPEG images, see Section V-D).

#### B. Problem of Traditional Approaches

As discussed in Section II-B, there are two main approaches for the dc coefficient recovery problems. In fact, for this special case that only dc coefficients are missing and all ac coefficients remain for each  $8 \times 8$  block in JPEG images, the relative pixel value distribution has accurately remained. On the contrary, if we use the method operated in the spatial domain like using the LP approach, even the transition between neighbor blocks becomes smoother, most of the ac coefficients are changed. Therefore, we not only failed to recover the accurate dc coefficient but also introduce errors for the ac coefficients. Thus, in this article, we use the first approach that is focused on how to recover the image content by accurately recovering the dc coefficients without changing the ac coefficients.

We indicate that the basic observed theory in [10] does not fit the practical scenario with an example shown in Fig. 3. These two  $8 \times 8$  blocks are picked from an image and the neighbor two



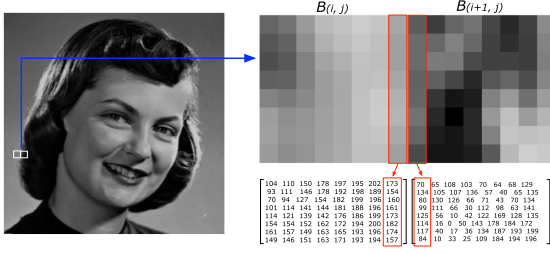


Fig. 3. Real case of pixel value distribution that does not fit the zero-mean Laplacian distributed variance between two neighbor blocks [10].

vectors of the neighbor pixels are very different. In this case, the method in [9] always tries to find the dc value to achieve the minimum mean square error (MSE) of two neighbor blocks. However, this will introduce wrong predictions for the dc value since the MSE of the real case should be very large due to the presence of edge.

In Fig. 3, if we know the dc value of the block  $B(i, j)$  is 80, based on the min MSE method in [9], the predicted dc value of the block  $B(i+1, j)$  is 82. The difference of the two neighbor vectors in Fig. 3 is  $[7, -13, -15, 3, 9, -1, 2, 19]$  while the real dc value of the block  $B(i+1, j)$  is 49 and the difference of the neighbor eight pixels is  $[103, 20, 80, 62, 48, 68, 57, 73]$ . The fact is that the pixel value distribution of these two neighbor blocks is not the zero-mean Laplacian distributed variate. On the opposite, the variance between these two blocks is very intensive. This is the reason why the image recovery methods based on dc recovery in [9], [10] will have the propagated errors and the block artifacts in the image. Thus, in this article, the problem to be solved is that for cases shown in Fig. 3, which does not fit the observed statistical theory in [10].

### C. Problem Description

Some other related research works have shown the possibility of deploying DL models to solve similar problems in the DCT related questions. One of the most popular issues is to use the DL model to enhance the image quality [18], which can also be seen as to remove the noise of the image [19].

The problem we are dealing with in this article is not about the image with the noise defined as the traditional situation since we have the exactly correct ac coefficients compared with the JPEG images, which means we do not have errors in the high-frequency domain compared with the original JPEG images. In other words, since we have all correct ac coefficients in the frequency domain and the JPEG image is the target we want to rebuild, there is no noise existed. Therefore, directly deploying the image denoising technique is not suitable for our scenario since the denoising techniques are mainly used to remove noises existed in the spatial domain corresponding to the errors of coefficients in the high-frequency bands which do not exist in our case.

The problem described in this article is to improve the pixel value distribution theory-based method as the first step and then use the DL model to solve the block artifacts which is limited by the theory of the traditional approaches. The DL model can

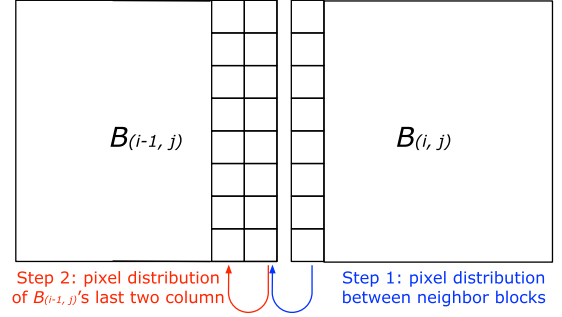


Fig. 4. Example of the improved DC recovery method proposed in this article: step 1: predict the dc coefficient of  $B(i, j)$  based on the pixel value distribution with the neighbor block  $B(i-1, j)$ ; step 2: predict the dc coefficient of  $B(i, j)$  based on the pixel value distribution trends of the last two columns of the neighbor block  $B(i-1, j)$ .

learn the compensate values between the original JPEG images and the images generated by the dc recovery methods. Thus, by using these two steps together as shown in Fig. 2, these two steps can recover the JPEG images with very limited information at the receiver's end. Therefore, we do not use the denoising DL models but deploy the deep residual learning model to realize our target that is to remove the block artifacts generated by the state-of-the-art dc recovery method for JPEG images.

## IV. DESIGN AND IMPLEMENTATION

In this section, we propose the designs and implementation details of our method including mainly two parts. The first part of our design is to overcome the error propagation issue by improving the method proposed in [9]. Then, we propose a deep CNN model, block artifact removing convolutional neural networks (BARCNN), to further enhance the recovery quality of JPEG images. The DL model formulation, architecture, and implementation details are presented.

### A. Improved DC Recovery Method

In this subsection, we propose the preprocess step in Fig. 2 based on improving the previous approach in [9]. According to our observation, the ratio that the pixel value distribution of the neighbor  $8 \times 8$  blocks fit the zero-mean Laplacian distributed variate theory in [10] is more than 90%. On one hand, for blocks that satisfy this condition, we improve the method in [9] to enhance the smoothness of pixel value distribution. We consider the pixel value distribution not only between two neighbor blocks but also within the blocks. As shown in Fig. 4, the first step is to calculate the MSE between adjacent pixels between two blocks. The second step is to calculate the difference between the last two columns of pixels within the neighbor block, which is used as the trend of the pixel value. By minimizing the gap between these two differences, we can get more accurate dc values.

On the other hand, for blocks that do not fit the zero-mean Laplacian condition, we improve the method in [9] by considering multiple directions of predicting the dc values. The motivation is that even if the blocks do not fit the zero-mean

**TABLE I**  
MAJOR NOTATIONS USED IN THE ALGORITHM AND THEIR DEFINITIONS

Notation	Definition
$B_{(i,j)}$	$8 \times 8$ block with location $(i, j)$
$A_{(i,j)}$	63 AC coefficients of block $B_{(i,j)}$
$D_{(i,j)}$	DC coefficient of block $B_{(i,j)}$
$P_{(i,j)}$	Pixel values of block $B_{(i,j)}$
$\tilde{D}_{(i,j)}$	Estimated DC coefficient of block $B_{(i,j)}$
$Q_{50}$	Quantization table of JPEG used in this paper
$v_n$	Number of $8 \times 8$ blocks in vertical direction
$h_n$	Number of $8 \times 8$ blocks in horizontal direction
$iDCT$	Inverse Discrete Cosine Transform
$Concat$	Concatenation of two arrays
$MSE$	MSE between two neighbor blocks defined in [9]

**Algorithm 1:** DC Coefficients Prediction From Upper-Left to Bottom-Right Corner.

**Input:** AC coefficients  $A_{(i,j)}$  with  $(i, j) \in \{v_n, h_n\}$ , DC coefficient  $D_{(1,1)}$  of upper-left block

**Output:** Recovered DC coefficients  $\tilde{D}_{(i,j)}$  with  $(i, j) \in \{v_n, h_n\}$

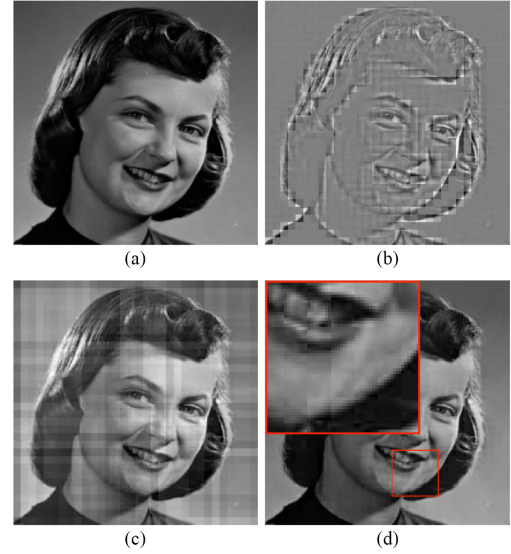
```

 $\tilde{D}_{(1,1)} = D_{(1,1)}$ 
for  $i \leftarrow 1$  to  $v_n$  do
  for  $j \leftarrow 1$  to  $h_n$  do
    /* Find the optimal DC that minimize loss. */
    for  $DC \leftarrow -64$  to  $64$  do
      /* Calculate spatial domain value. */
       $P(i, j) = iDCT(Concat(DC, A_{(i,j)}) * Q_{50})$ 
       $P(i-1, j) = iDCT(Concat(\tilde{D}_{(i-1,j)}, A_{(i-1,j)}) * Q_{50})$ 
       $P(i, j-1) = iDCT(Concat(\tilde{D}_{(i,j-1)}, A_{(i,j-1)}) * Q_{50})$ 
       $MSE(DC) = MSE(P_{(i,j)}, P_{(i-1,j)}) + MSE(P_{(i,j)}, P_{(i,j-1)})$ 
    end for
     $\tilde{D}_{(i,j)} = \arg \min_{DC} \{MSE(DC)\}$ 
  end for
end for

```

Laplacian condition with neighbor blocks in one direction, it is still possible that this condition is satisfied in other directions. For instance, we can deploy the scheme in Fig. 4 from the upper-left to bottom-right corner of the image, as shown in Algorithm I where the notation is defined in Table I. The dc value in the upper-left corner should be preserved and used as the reference value. Then, the unknown dc values can be predicted block by block by using upper and left neighbor blocks. Although the prediction errors can be reduced by our method, the propagation of errors still exist. To correct the errors, we further deploy the same scheme from the other three corners, i.e., from upper-right to bottom-left, from bottom-left to upper-right, and from bottom-right to upper-left, respectively. Finally, we calculate the average dc values of these four processes as the predicted values.

The results of this design are shown in Fig. 5. There are still some block artifacts that existed if we look into details of the image content [e.g., see the Fig. 5 (d)] although we could observe a clear improvement compared with the previous method in [9].



**Fig. 5.** Results of the improved dc prediction method proposed in this article: (a) Initial JPEG image. (b) JPEG compressed image with no dc coefficients. (c) Recovered image with dc predicted based on methods in [9]. (d) Recovered image with dc predicted in this article.

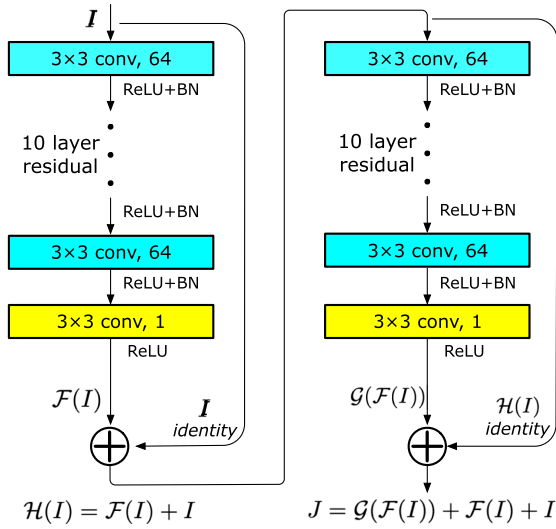
The PSNR of the recovery result in Fig. 5 (d) compared with the JPEG image is around 25 dB. The improvement in visual effects is very obvious but there are still prediction errors leading to the block artifacts.

The fundamental reason for block artifacts in dc prediction is that the zero-mean Laplacian condition in [10] that we relied on is not always true in the case of real-world images. This fact leads to the limitation that no matter how we try to use the average prediction methods to make up the errors of dc coefficients, there are always some blocks that do not fit the zero-mean Laplacian condition. Therefore, the next question is that if we can summarize a statistical model to correct the errors in the dc recovery with a large number of images. We answer this question by using the DL models to explore the real-world trends of the pixel value distribution for correcting the errors in dc prediction.

### B. DL Model Formulation

BARCNN is used to improve the image quality of the recovered image of the preprocess step in Fig. 5. The purpose is to learn mapping functions between the input image  $I$  recovered by the method in Section IV-A and the original image  $J$ . For achieving this purpose, we improved the model in [20] by combining the deep residual learning framework [12] to design our model. The input image of our model can be considered as  $I = J + N$ , where  $J$  is the original JPEG image and  $N$  is the image representing block artifacts caused by propagation error mentioned in Section II. We aim to learn a residual mapping function  $\mathcal{R}(I)$  that satisfies  $\mathcal{R}(I) \approx N$ . Thus, the loss function can be formulated by the averaged MSE as follows:

$$\mathcal{L}(\Theta) = \frac{1}{N} \sum_{i=1}^N \|\mathcal{R}(I_i; \Theta) - (J_i - I_i)\|_F^2 \quad (1)$$



**Fig. 6.** Architecture of the proposed BARCNN model, which is inspired by residual learning framework [12]. There are two residual blocks used to learn the residual mapping between the input image and noise image that represents block artifacts. Inside each block, there are 12 convolutional layers with ReLU activation. Batch normalization is adopted before activation. More details are in Section IV-C.

where  $\theta$  is the trainable parameters of our model,  $N$  is the number of training images.  $I_i$  and  $J_i$  are the  $i$ th corresponding input image and the original image.  $\|\cdot\|_F$  denotes the Frobenius norm. As illustrated in Fig. 6, there two residual blocks in the BARCNN model. The first block learns a residual mapping  $\mathcal{F}(I) = \mathcal{H}(I) - I$ . The second block learns another residual mapping  $\mathcal{G}(\mathcal{H}(I)) = \mathcal{K}(I) - \mathcal{H}(I)$ . Since  $\mathcal{K}(I)$  is the approximation of the original image  $J$ , then the objective residual mapping  $\mathcal{R}(I) = \mathcal{G}(\mathcal{H}(I)) + \mathcal{F}(I)$ . Hence, the loss function in (1) can be further formulated as

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \|\mathcal{G}(\mathcal{H}(I_i); \Theta) + \mathcal{F}(I_i; \Theta) - (J_i - I_i)\|_F^2 \quad (2)$$

By using residual learning, our model can learn a residual mapping by a combination of several nonlinear mappings. This approach makes it powerful to learn more complex features for image compression artifacts.

### C. DL Model Architecture

Instead of using only one residual mapping in [20], we improve the previous work by using two residual blocks to obtain a deeper network with better performance. As illustrated in Fig. 6, two blocks have identical architecture while each residual block contains 12 convolutional layers and one shortcut connection. In a block, all convolutional layers have 64 filters of size  $3 \times 3$  with a stride of one, except from the last one that has only one filter of size  $3 \times 3$  filters with a stride of one, which makes the output have the same shape as the input. All layers use rectified linear units (ReLU) as the activation function to produce nonlinearity. The batch normalization is also used from the second layer to the eleventh layer. According to [21], increasing network depth using an architecture with very small convolutional filters can

significantly improve the performance of the model. Thus, we use filters of size  $3 \times 3$  and build a neural network with a depth of 24. The important features in our model are the use of residual blocks and batch normalization, which make the network easier to be optimized and obtain promising results. The use of residual learning follows the research direction in [12].

In our case, the noise image  $N$  represents block artifacts caused by error propagation in the preprocessed dc recovery. Compared with the original image  $J$ ,  $N = J - I$  has relatively smaller values. It would be easier to optimize a mapping that fits  $N$  rather than  $J$ . Hence, nonlinear functions  $\mathcal{F}$  and  $\mathcal{G}$  are trained to fit the residual mappings. In each residual block, there is one shortcut connection, which represents an identity mapping. According to [12], this kind of shortcut connection helps to solve the degradation problem that appears in simply stacked nonlinear layers. With this architecture, we obtain a deeper CNN model that has higher complexity while is still easy to train. The experiment results show that this architecture is effective for our removing block artifacts task.

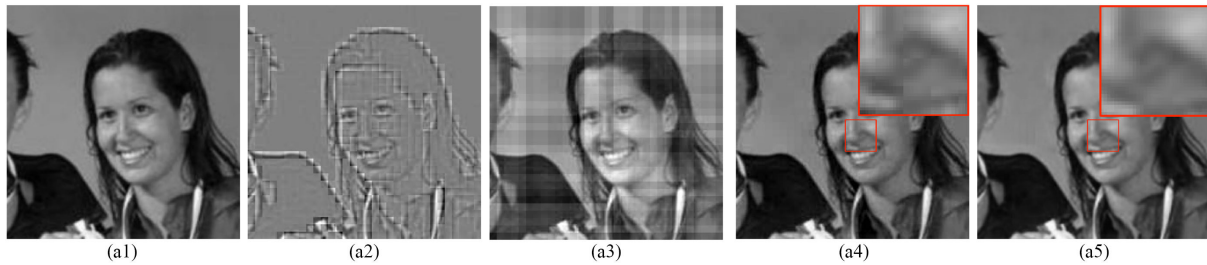
The use of batch normalization is inspired by [22]. The change in the distributions of internal nodes of a deep neural network, so-called “internal covariate shift,” is considered to having a negative effect on training efficiency. In [23], the actual impact of batch normalization is to make the optimization landscape significantly smoother. This smoothness can result in more efficient training, which motivates us to use this mechanism in our model. BARCNN model is trained by using the minibatch Adam optimization algorithm. Thus, normalization can be adopted for each minibatch during each iteration. To stabilize the distribution of inputs to the nonlinear function, the batch normalization is performed before ReLU activation in each layer.

### D. DL Model Implementation

To train the BARCNN model, we apply a data augmentation method by using cropped image patches. The input image  $I$  of our model is a cropped patch of the image recovered by the method in Section IV-A. The size of the patch should be well chosen to contain significant patterns. Since the DCT transform is performed block by block with a size of  $8 \times 8$ , there will be block artifacts on the boundaries between different blocks. Moreover, errors in predicting dc values result in an intensity difference. To better extract these artifacts, we set the size of the patch as 32 to have 16 DCT blocks in one cropped image. The experiment results show that patches that we generated contain enough information for removing artifacts.

The cropped images are then fed into the BARCNN model. Convolutional layers use filters to extract high-level features in the image. Batch normalization helps to stabilize the distribution of those features. And the ReLU activation ensures the nonlinearity of each layer. After 12 layers, the output of the first residual block has the same size as the input, which can be considered as the input image plus the related noise image. Then, the second residual block is used to further tune the noise image through the same procedure. The final output is the enhanced image of  $J$ , which will be compared with the reference cropped image through loss function (2). Adam optimization algorithm is used





**Fig. 7.** Comparison of one example image. (a1) Original JPEG image. (a2) JPEG compressed image without dc (PSNR: 12.59 dB, SSIM: 0.5285, compression: 63.58%). (a3) Recovered image by method in [9] (PSNR: 17.91 dB, SSIM: 0.7978). (a4) Recovered image by preprocess step (PSNR: 29.44 dB, SSIM: 0.9592). (a5) Further enhanced image by BARCNN model (PSNR: 32.63 dB, SSIM: 0.9652).

to optimize the loss with an adaptive learning rate. More details about settings during the training procedure will be introduced in Section V-A.

## V. EXPERIMENTATION AND EVALUATION

In this section, we first list the key implementation details of the BARCNN model training. Then, the result evaluation including the visual effects analysis and the statistical analysis is proposed. The compression ratio of the system proposed in Fig. 2 is also calculated on several famous data sets with images of different sizes and properties to evaluate the efficiency of our method. At the end of this section, we also give a use case analysis to illustrate how our method can be deployed in an IoT system.

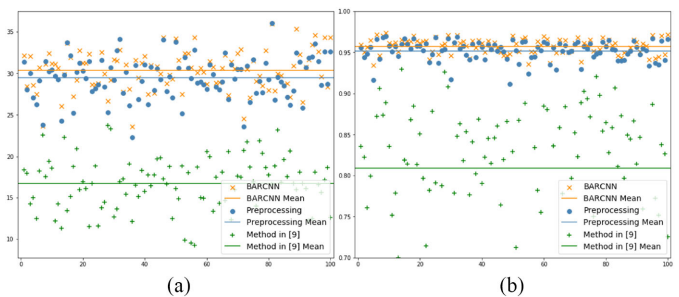
### A. Data Preparation and Model Training

We use image data sets from the labelled faces in the wild (LFW) in [24]. In total, 5000 human face images are used as a data set: 4000 out of the 5000 images are used as a training data set and the rest of 1000 as a testing data set. We apply JPEG compression to all images and set all dc coefficients to zero except the four dc coefficients in corners. Then, we use the preprocessing step shown in Section III to predict the dc coefficients and the images with block artifacts are get. These images are used as the training data set by cropping each image into 225 patches of size  $32 \times 32$  with a stride of 16. Thus, the training data have 900 000 cropped images and the testing data set has 225 000 cropped images. The reference data have the same number of images cropped from original images. All pixel values in images are normalized to values in  $[0, 1]$ .

We use Keras package [25] with Tensorflow [26] backend to implement our model. Weights in all convolutional layers are initialized by a random generated orthogonal matrix. The hyper-parameters of Adam optimizer are set by  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ . The initial learning rate is 0.0001. Experiments are done on a platform with Intel(R) Xeon(R) CPU E5-2620v3 @ 2.40 GHz CPU and three NVIDIA Tesla K80 GPUs. The minibatch size is 256 for each GPU. The loss function converges after about 50 epochs.

### B. Visual Evaluation

As pointed in Section V-A, we tested 1000 images in the LFW data set to measure the effectiveness of the proposed BARCNN



**Fig. 8.** Statistical results comparison between the images recovered by methods in [9], by preprocess step in this article, and by BARCNN model on preprocessed images: (a) PSNR. (b) SSIM.

model. In Fig. 8, we picked one example from the testing data set to show the visual results. We compare the visual effects of the recovered image by our methods with the original JPEG image, the image with dc coefficients being zeros, and the recovered image by method in [9]. As we can observe, for the image where all dc coefficients equal to zeros, there are only some blur edges of the original image but all gray scales are lost. For the improved method based on [9], the basic grayscale gradient is recovered but there are block artifacts due to the errors of the dc values. These errors on the dc values are mainly due to the error propagation in the dc guessing process pointed by [9], [10] which cannot be overcome by the traditional methods. Then, we can observe an obvious improvement based on the BARCNN model proposed in this article on the details. The block artifacts including the sharp edges are removed and the image quality is improved.

### C. Statistical Evaluation

We use PSNR and structural similarity (SSIM) [27] to measure the results. Note that PSNR is normally used to measure the effect of the noise compared with the ground truth images, here the PSNR is measured compared with the JPEG image for two images recovered in two steps as shown in Fig. 2. We improved the method in [9] to get a preprocess recovered image as shown in Fig. 7, which has block artifacts. Then, the proposed BARCNN model is used to further recover the preprocessed images to test the improvement on the PSNR value. In Fig. 8(a), we list the PSNR values of 100 images randomly picked from the test data set and plot the PSNR values of the

**TABLE II**  
COMPRESSION RATIO TESTS ON FIVE DATA SETS

	LFW100	BSDS100	General100	Urban100	Manga109
min	54.93%	44.19%	42.59%	46.33%	49.54%
max	70.37%	74.25%	78.05%	82.82%	81.13%
ave	63.84%	63.94%	63.59%	66.50%	66.38%

preprocess recovered images and the recovered images based on BARCNN model. The average PSNR values are also calculated, respectively. We can observe an obvious improvement of the PSNR values where the average PSNR values are 29.2 and 30.3 dB, respectively. SSIM is used to test image quality after transmission and recovery. For the preprocess recovered images, the average SSIM can reach 0.95 which is the best result in the traditional block-based approach. The average SSIM of BARCNN reaches 0.96, which is even better. Moreover, for the preprocess recovered images, the SSIM could be relatively lower until 0.91 while the DL-based method is more than 0.93.

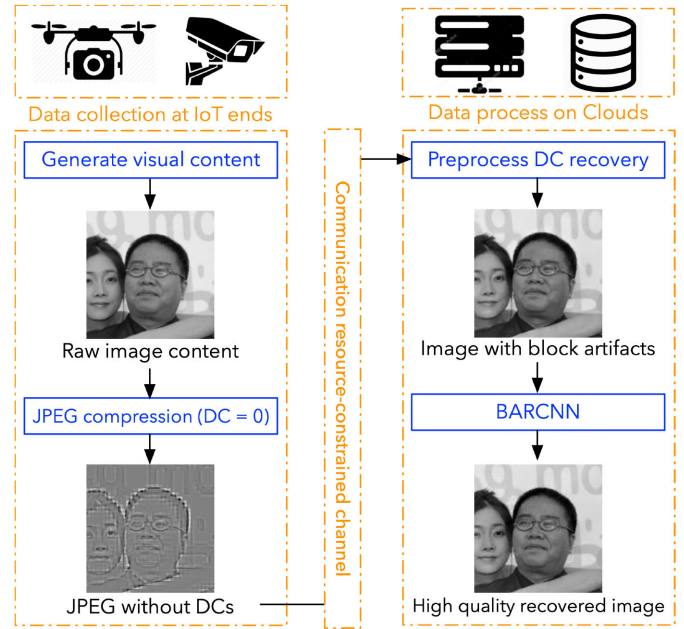
### D. Compression Ratio Evaluation

The baseline scenario is that the sender will send the JPEG images with a standard JPEG compression procedure such that the receiver will get a JPEG image. The second scenario, as shown in Fig. 2, uses the same procedure to generate a JPEG image with replacing all dc coefficients as zeros except the four dc coefficients in the four corner blocks.

For the proposed scenario, we test the compression ratio on five benchmark data set. The compression ratio is calculated by the ratio between JPEG images with only four dc coefficients and original JPEG images. The results are shown in Table II. LFW100 contains 100 images selected from our testing data set. Four others are data sets used in [28]. Some images have a compression ratio of less than 50%, which means that they can save half of the storage of JPEG images. The average compression ratio is around 65%. The high compression ratio shows the effectiveness of the proposed JPEG transmission procedure. Noticed that in practice, several factors influence the compression ratio, image size, average intensity, spectral distribution, etc.

### E. Use Case Analysis in IoT Systems

In this subsection, we will present a use case for deploying our methods in the IoT system for image transmission. Nowadays, with the development of DL technologies, the growth of the IoT is rapid. Equipped with different image collection sensors, IoT devices become appealing targets for DL applications. However, an IoT system normally consists of a large number of edge devices with high-quality sensors streaming information at a very high-data rate (e.g., video surveillance [29], remote sensing [30]). This will generate a large amount of data transferring between the sensor nodes and the central server, which will lead to communication needs. However, the communication resources in IoT systems are normally constrained is low which needs the IoT ends to compress the images and the server ends could recover. Additionally, due to the limited transmission energy budget, the compression operation on IoT ends must be low energy cost.



**Fig. 9.** Use case of deploying our method in the IoT system. 1) Little extra cost compression algorithm at IoT ends. 2) Effective image recovery with two steps at server ends.

The use case is illustrated in Fig. 9. Since our method uses the same JPEG compression procedure but only changes all dc values to zeros, there are almost no extra computing costs on the IoT ends. Then, the size of one compressed image is only 60% of a normal JPEG image, which can significantly reduce the transmission bandwidth needs by around 40%. In the end, at the server end, the two-step method could effectively reconstruct the high-quality image with a PSNR more than 30 dB.

## VI. DISCUSSION AND FUTURE WORK

In this article, the research motivation is to answer the question that whether the dc coefficients can be accurately recovered if the ac coefficients are exactly kept as in [10]. The initial research is based on improving the existing approaches but the limitation is very obvious since the basic theory is not 100% true in real-world cases. Although the PSNR measurement is acceptable the block artifacts are very obvious, which became an obstacle that cannot be overcome by the traditional methods. Then, we redefined this obstacle as a question of building a model to fit the pixel value distribution property of real-world images which inspired us to deploy the DL model. The method in this article is to combine the improved methods for accurately dc coefficients recovery, which is the state-of-the-art method, with the deep residual learning model. By recovering dc coefficients with acceptable accuracy, the deep residual learning model can be used to finely tune the pixel values to achieve a block-free image with higher PSNR.

The experimentation showed that the average PSNR compared with the JPEG is more than 30 dB and the transmission data ratio is only around 60% of the initial JPEG image. Moreover, considering a use case like IoT [31], this method does not require any additional operation on the sender's end but can reduce 40% of data to be transmitted. Thus, we believe this research



has achieved our initial target that could highly improve the transmission efficiency for DCT-based multimedia big data like JPEG images.

On the one hand, we are currently exploring the possibility of enhancing the JPEG image compression by recovering the dc coefficients at the receiver's end. On the other hand, such transformation also provides an idea for designing SE methods [32]. Since the transformation has already provided a choice for defining different importance levels for the frequency coefficients, SE methods can protect the image in a lightweight manner by encrypting only dc coefficients [33]. Such protection is also proved to be insecure by our work since we have proved the possibility to recover the dc coefficients for DCT-based SE methods on multimedia data.

For future work, we will further explore better DL models to improve the recovery accuracy of the recovered images with more data sets. Also, the calculation cost for performing image recovery will be considered in a real-world scenario to explore the feasibility of using this method at the receiver's end.

## VII. CONCLUSION

In this article, we proposed an enhanced JPEG compression method by recovering the image from only four dc coefficients at the receiver's end. We first proposed a state-of-the-art accurate dc recovery method as the preprocess step to generate an image with only the ac coefficients and four dc coefficients. Then, in order to get rid of the problem that the observed theory cannot fit all real-world image property, we proposed a deep residual learning model to further remove the block artifacts based on the results of the preprocessed images. The experimentation showed that transmitting only four dc coefficients and all ac coefficients will take only around 60% data of the original JPEG image while the recovery method can generate an image with average PSNR more than 30 dB for many different JPEG image datasets. We believe the proposed method could help to significantly improve the efficiency of DCT-based multimedia big data transmission in IoT scenarios since we reduce 40% of data to be transmitted without any additional computing tasks on the IoT devices.

## REFERENCES

- [1] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain intelligence: Go beyond artificial intelligence," *Mobile Netw. Appl.*, vol. 23, no. 2, pp. 368–375, 2018.
- [2] F. Tang, L. Barolli, and J. Li, "A joint design for distributed stable routing and channel assignment over multihop and multiframe mobile ad hoc cognitive networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1606–1615, May 2012.
- [3] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
- [4] J. Lainema, M. M. Hannuksela, V. K. M. Vadakital, and E. B. Aksu, "HEVC still image coding and high efficiency image file format," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 71–75.
- [5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
- [6] J. Zeng, O. C. Au, W. Dai, Y. Kong, L. Jia, and W. Zhu, "A tutorial on image/video coding standards," in *Proc. IEEE Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2013, pp. 1–7.
- [7] Q. Zhang, T. Huang, Y. Zhu, and M. Qiu, "A case study of sensor data collection and analysis in smart city: Provenance in smart food supply chain," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 11, 2013, Art. no. 382132.
- [8] C. Chen, Z. Miao, X. Meng, S. Zhu, and B. Zeng, "DC coefficient estimation of intra-predicted residuals in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1906–1919, Aug. 2018.
- [9] H. Qiu, G. Memmi, X. Chen, and J. Xiong, "DC coefficient recovery for JPEG images in ubiquitous communication systems," *Future Gener. Comput. Syst.*, vol. 96, pp. 23–31, 2019.
- [10] T. Uehara, R. Safavi-Naini, and P. Ogunbona, "Recovering DC coefficients in block-based DCT," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3592–3596, Nov. 2006.
- [11] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 576–584.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [13] G. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.
- [14] R. Kresch and N. Merhav, "Fast DCT domain filtering using the DCT and the DST," *IEEE Trans. Image Process.*, vol. 8, no. 6, pp. 821–833, Jun. 1999.
- [15] S. Li, A. Karrenbauer, D. Saupe, and C.-C. J. Kuo, "Recovering missing coefficients in DCT-transformed images," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 1537–1540.
- [16] H. Qiu, N. Enfrin, and G. Memmi, "A case study for practical issues of DCT based bitmap selective encryption methods," in *Proc. IEEE Int. Conf. Secur. Smart Cities, Ind. Control Syst. Commun.*, 2018, pp. 1–7.
- [17] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. Berlin, Germany: Springer, 1992.
- [18] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9446–9454.
- [19] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3929–3938.
- [20] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. *arXiv:1502.03167*. [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [23] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2483–2493.
- [24] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts, Amherst, MA, USA, Oct. 2007.
- [25] N. Ketkar, "Introduction to Keras," in *Deep Learning With Python*. Berlin, Germany: Springer, 2017, pp. 97–111.
- [26] M. Abadi et al., "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operat. Syst. Design Implementation*, 2016, pp. 265–283.
- [27] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2010, pp. 2366–2369.
- [28] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 624–632.
- [29] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," in *Proc. Eur. Conf. Comput. Vis.* 2016, pp. 869–884.
- [30] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.
- [31] H. Su, M. Qiu, and H. Wang, "Secure wireless communication system for smart grid with rechargeable electric vehicles," *IEEE Commun. Mag.*, vol. 50, no. 8, pp. 62–68, Aug. 2012.
- [32] H. Qiu, H. Noura, M. Qiu, Z. Ming, and G. Memmi, "A user-centric data protection method for cloud storage based on invertible DWT," *IEEE Trans. Cloud Comput.*, to be published, doi: [10.1109/TCC.2019.2911679](https://doi.org/10.1109/TCC.2019.2911679).
- [33] L. Krikor, S. Baba, T. Arif, and Z. Shaaban, "Image encryption using DCT and stream cipher," *Eur. J. Sci. Res.*, vol. 32, no. 1, pp. 47–57, 2009.



**Han Qiu** (Member, IEEE) received the B.E. in communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2007, the M.S. degree in network and security from Institute Eurecom, Biot, France, in 2013, and the Ph.D. degree in computer science from the Department of Networks and Computer Science, Telecom-ParisTech, Paris, France, in 2017.

He is currently a Post-Doctoral Researcher with the Department of Network and Computer Science, Telecom-ParisTech, Paris, France. His research interests include heterogeneous computing, cybersecurity, applied cryptography, multimedia security.



**Qinkai Zheng** received the bachelor's degree in information engineering from the SPEIT, Shanghai Jiao Tong University, Shanghai, China, in 2018. He is currently working toward the master's degree majored in computer vision in a double degree program between Shanghai Jiao Tong University and Telecom Paris, Paris, France.

His research interest includes machine learning security in computer vision.



**Gerard Memmi** (Member, IEEE) received the Ph.D. (These d'Etat) degree in computer science from Universite Pierre et Marie Curie, Paris, France, in 1983.

He has been a Professor and Head of the Networks and Computer Science Department, Telecom-ParisTech, Paris, France, since 2009. He is member of the executive board of the IRT SystemX since 2012. Before joining Telecom-ParisTech, he held various executive positions in American start-ups. He succeeded in delivering

the industry best-in-class equivalency checker used to verify electronic design; and focused on improving its architecture and performances. While founding and developing the Applied Research Laboratory for Groupe Bull in the US, he was honored as Principal Investigator for a DARPA grant on Collaborative Software. He has authored or coauthored more than 90 publications including patents, co-authored a book, gave key notes presentations in international conferences. He is holding a these d'Etat in Computer Science from Universite Pierre et Marie Curie, Paris. He is constantly involved in the development of key scientific and industrial partnership. His current research interests are data protection and privacy, energy profiling of software programs, and verification of distributed systems.



**Jiali Lu** received the M.S. and the M.E. degrees in computer science with honor from the Department of Telecommunication, INSA Lyon, France, in 2004, where he also received the Ph.D. degree in computer science, in 2008.

He is an Associate Professor and an Assistant Dean with ParisTech, Paris, France and Shanghai Jiao Tong University, Shanghai, China, and a Researcher with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He has authored or co-authored more than 50 publications in international journal and conferences in this area. His current research interests include wireless networks, vehicle networks, and security aspects of machine learning.



**Meikang Qiu** (Senior Member, IEEE) received the B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1992 and 1998, respectively, and the Ph.D. degree from the University of Texas at Dallas, Richardson, TX, USA, in 2007, all in computer science.

He is the Department Head and tenured Full Professor of Texas A&M University Commerce, Commerce, TX, USA. He has authored or co-authored more than 20 books, more than 550 peer-reviewed journal, and conference papers,

including more than 80 IEEE/ACM Transactions papers. His current research interests include cyber security, big data analysis, cloud computing, smarting computing, intelligent data, embedded systems, etc.

Dr. Qiu is ACM Distinguished Member. He is the Chair of IEEE Smart Computing Technical Committee. He is an Associate Editor for 10+ international journals, including IEEE TRANSACTIONS ON COMPUTERS and IEEE TRANSACTIONS ON CLOUD COMPUTING.



**Bhavani Thuraisingham** (Fellow, IEEE) received the Ph.D. degree in theory of computation from the University of Wales, Swansea, U.K., in 1979, and higher doctorate (D.Eng) degree in secure data management from the University of Bristol, Bristol, U.K., in 2011

She is the Founders Chair Professor of Computer Science and the Executive Director of the Cyber Security Research and Education Institute with the University of Texas at Dallas, Richardson, TX, USA. She has worked in industry (Honeywell), federal laboratory (MITRE), US government (NSF) and her work has resulted in 130 journal articles, 300 conference papers, 150 keynote and featured addresses, six US patents, 15 books as well as technology transfer to operational systems and commercial products.

Prof. Thuraisingham is also a visiting Senior Research Fellow with Kings College, University of London and a Fellow of the ACM and the AAAS. She was the recipient of several awards including the IEEE Computer Society 1997 Technical Achievement Award, ACM Special Interest Group on Security, Audit and Control 2010 Outstanding Contributions Award, and the ACM Symposium on Access Control Models and Technologies 10 Year Test of Time Award in 2018 and 2019. She also Co-Chaired the Women in Cyber Security Conference in 2016 and delivered a featured address at Stanford University Women in Data Science Conference in 2018.

Prof. Thuraisingham is also a visiting Senior Research Fellow with Kings College, University of London and a Fellow of the ACM and the AAAS. She was the recipient of several awards including the IEEE Computer Society 1997 Technical Achievement Award, ACM Special Interest Group on Security, Audit and Control 2010 Outstanding Contributions Award, and the ACM Symposium on Access Control Models and Technologies 10 Year Test of Time Award in 2018 and 2019. She also Co-Chaired the Women in Cyber Security Conference in 2016 and delivered a featured address at Stanford University Women in Data Science Conference in 2018.