IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 29, 2020, accepted January 11, 2021, date of publication January 14, 2021, date of current version January 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051678

End-to-End Anti-Forensics Network of Single and Double JPEG Detection

DOHYUN KIM[1], WONHYUK AHN[1], AND HEUNG-KYU LEE[1,2]
[1]School of Computing, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
[2]Digital Innotech Company, Daejeon 34184, South Korea

Corresponding author: Heung-Kyu Lee (heunglee@kaist.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant NRF-2019R1A2C2084569.

ABSTRACT JPEG compression is one of the major image compression methods and is widely used on the Internet. In addition, identifying traces of JPEG compression and double JPEG compression (DJPEG) is crucial in the image forensics field. Therefore, JPEG compression detection and DJPEG compression detection are two of the popular image authentication methods. Many feature-based JPEG detection methods have been proposed for that purpose, and there have been outstanding improvements in DJPEG detection with the development of deep learning. A number of anti-forensics of JPEG detection that counter feature-based detectors have been proposed but only a few techniques that counter DJPEG have been researched. This paper explores whether JPEG reconstruction methods, including restoration and anti-forensics of JPEG detection, can deceive JPEG and DJPEG detectors. We demonstrate that existing anti-forensics of JPEG detection can deceive both JPEG and DJPEG detectors well but perform poorly in non-aligned cases and degrade the image quality. We propose a convolutional neural network (CNN) based anti-forensics method to improve the performance of anti-forensics so that they can proficiently deceive JPEG and DJPEG detectors with higher image quality. Moreover, we explore the generalization algorithm to handle the real scenario.

INDEX TERMS Anti-forensics, CNN, DJPEG detection, image forensics, JPEG detection, JPEG restoration.

I. INTRODUCTION

Through the development of the IT industry, computers, and smartphones have become essential parts of our lives, and almost all data, including images, are now stored in digital form. People exchange images and chat online about them. Meanwhile, many images undergo various types of image processing, which instantiates a large difference between the processed image and the original. Image editing tools such as *Photoshop* and editing smartphone apps have made it easier for people to manipulate images nowadays. Moreover, deepfakes and style-transfers, which have been extensively developed through the application of deep learning, help people edit images more effectively. Because of the impossibility of identifying manipulated images with our eyes, they can easily be abused. To tackle these issues, image forensics has been widely researched. It aims at verifying the authenticity of digital images without any signatures or watermarks. It usually investigates footprints left by copy-moves [1], splicing [2], retouching [3], deepfakes [4], etc.

On the other hand, the majority of the images on the Internet are encoded into JPEG format because of its compression effectiveness. As JPEG compression is a lossy compression, it leaves strong traces that can be used in forensics. Specifically, JPEG detection can be applied for identifying JPEG artifacts in uncompressed or losslessly compressed file images [5]. Additionally, a history of recompression can signify the presence of an abnormal image because it indicates that tampered images have been resaved. Therefore, many research studies on JPEG detection and double JPEG (DJPEG) detection have been conducted.

Attackers who do not want to be exposed to JPEG and DJPEG detectors decompress JPEG images and manipulate them using several algorithms, as shown in Fig. 1. In doing so, attackers must not only ensure that an image follows a single JPEG distribution statistically but also reconstruct images to have an invisible damaged region.

Anti-forensics, which are designed to mislead forensic investigators, can help researchers study the vulnerabilities

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

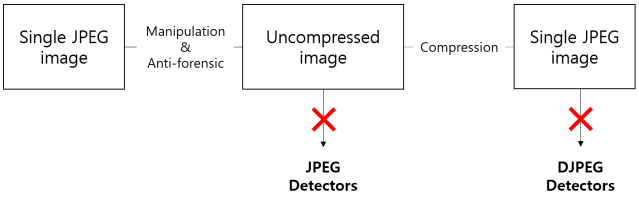

FIGURE 1. A scenario in which attackers deceive JPEG and DJPEG detectors. The anti-forensical operation in the JPEG image disguises the uncompressed image and was able to deceive JPEG detectors. Recompression of the processed images could be approximated to the single JPEG and could deceive DJPEG detectors.

of existing forensic techniques to further develop trustworthy digital forensics against attackers [6], [7]. Accordingly, applying anti-forensics of JPEG detection, which transform image statistics to mislead the detectors into classifying JPEGs as uncompressed, have been proposed [8]–[10]. In contrast, although DJPEG detection methods developed through deep learning have shown promising results, there are no previous anti-forensics analyses that target those algorithms. Therefore, researchers must investigate the weaknesses of existing methods and devise a more reliable DJPEG detector.

Moreover, due to the degradation of the visual quality caused by JPEG compression, JPEG restoration tasks have been proposed [11]–[13], to restore JPEGs to their uncompressed versions, which have higher visual quality. JPEG restoration research studies have shown competitive results in visual quality using deep learning, but none of them have demonstrated that they can actually remove JPEG artifacts [14]. However, it is plausible that JPEG restoration could be one of the JPEG manipulation tasks that used by attackers to remove the artifacts.

In this paper, we conduct experiments with both anti-forensics of JPEG detection and JPEG restoration tasks to determine whether they erase JPEG compression traces in the cases of the JPEG and DJPEG domains. Moreover, the previous state-of-the-art anti-forensics method [10] is based on multi-step subgradient optimization method, which is time consuming. Furthermore, the CNN-based anti-forensics study using a Generative Adversarial Network (GAN) was studied but the researchers did not analyze the detectability to DJPEG detectors and showed less undetectability [15]. In contrast to the earlier studies, we propose a convolutional neural network (CNN) that generates higher-quality images fast and can competently deceive JPEG and DJPEG detectors. We propose the use of anti-forensics loss in the anti-forensics training for removing JPEG traces, and we adopt the EDSR network [16], which is simple and high-performing in super-resolution, for visual quality.

We train and evaluate the proposed method with the BossBase 1.01 [17] and BOWS2 [18] datasets. We use the detection accuracy rate and minimum decision error for evaluating the undetectability and for the visual quality metrics, we use the peak-signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). The contributions of our paper are summarized as follows.

- We analyze the effectiveness of applying JPEG restoration and JPEG anti-forensics methods to CNN-based DJPEG detectors.
- We propose a deep learning based end-to-end anti-forensics with anti-forensical loss functions that targets to both JPEG and DJPEG.
- We show that our trained network provides high undetectability against JPEG and DJPEG detectors and achieves a higher visual quality than previous works.

II. RELATED WORKS

A. JPEG DETECTION AND ANTI-FORENSICS

Many researchers have developed JPEG compression detection techniques by discovering statistical differences between uncompressed and JPEG images. Fan *et al.* [5] measured a blocking signature that compares the difference in the distribution of neighboring pixels at the boundary and the center. Besides, they proposed a maximum-likelihood estimation of the quantization table and used it for detection by counting the number of estimated quantization tables that are larger than 1. Luo *et al.* [19] proposed an algorithm using the difference of the AC coefficients in the range $(-1, 1)$ and in the sum of the $(-2, -1)$ and $(1, 2)$ range. In the paper of Lai *et al.* [20], they extracted a calibration feature for detecting JPEG compression, which is an idea originating from the research of Fridrich *et al.* [21]. They computed the variances of the high-frequency discrete cosine transform (DCT) in both the original and calibrated images, which was cropped by four pixels both horizontally and vertically, and used the difference of each variance for the detection. Valenzise *et al.* [22] used the difference of the maximum total variation in two images that were recompressed with sequential quality factors (QFs). Fan *et al.* [23] detected JPEG compression by recording the difference of the gradient at the boundary and the center. Inter and intra block statistics and the subtractive pixel adjacency matrix (SPAM) feature [24], [25] were used as a feature of support vector machine (SVM) [22], [26].

The effectiveness of anti-forensics of JPEG detection has been studied through the statistical modification of the image. Based on the knowledge that the DCT distribution in the AC component can be modeled as a Laplacian distribution [27], Stamm *et al.* [8] added dithering noise to the DCT histogram that resembled the uncompressed image's DCT distribution. It deceived the quantization table estimation detector, but it degraded the visual quality of images and was detected through block measuring [5]. In [28], Stamm *et al.* applied median filtering after the dithering noise to counter block measuring. Valenzise *et al.* [9] proposed a perceptual anti-forensic dithering operation that achieved a higher visual quality than [8]. Fan *et al.* [23] proposed a constraint subgradient method that minimized the total variation, aiming to smooth and match the boundary and center distribution.

Later, Fan *et al.* [10] improved the visual quality of the image by applying a multi-step deblocking method, including TV minimization, perceptual histogram smoothing, and decalibration. This approach achieved a higher undetectability in JPEG compression detection and higher visual quality than previous attempts. However, their technique, the multi-step subgradient method, took an excessive amount of time to process. Singh *et al.* [29] diversified some of the steps of [10] to achieve a higher visual quality, but their method had some limitations regarding undetectability. Many algorithms for programming anti-forensics of JPEG detection have been proposed, and they have achieved high undetectability, but they all degrade visual quality much.

A CNN based anti-forensics of JPEG has been studied [15]. They used a GAN network, and the discriminator was designed as a CNN-based JPEG detector by adding a high pass filter in front of the network. They increased the visual quality, but they evaluated only one JPEG detector, and showed low undetectability even though they did not compare with the SOTA of anti-forensics of JPEG.

B. DJPEG DETECTION AND ANTI-FORENSICS

In the image forensics field, it is well known that the DJPEG compression leaves traces, particularly in the DCT domain. Therefore, most of the DJPEG detection methods rely on the statistical features of DCT coefficients. To give examples, Li *et al.* [30] presented a DJPEG detector that uses the first digits of the DCT coefficients as features. Lin *et al* [31] showed a DJPEG detector that could be used to localize forged regions in images.

With the development of deep learning, the performance of DJPEG detectors has significantly improved. Wang *et al.* [32] firstly proposed a CNN-based DJPEG detector that used a 1D histogram vector to judge the number of compressions. Barni *et al.* [33] improved upon this idea greatly by using a 2D histogram as an input. Park *et al.* [34] investigated real-world QFs and presented a more practical scenario where 1,120 quantization matrices existed. The authors optimized the network architecture and concatenated the quantization matrix into the last three fully connected layers to boost performance. The authors in [35] first proposed an end-to-end neural 3D CNN without having to manually generate a histogram. It improved the previous methods' ability to utilize raw DCT coefficients. The CNN-based DJPEG methods are all processed in the DCT domain, and most of them are processed in the histogram because their footprints remain in the DCT domain rather than the pixel domain.

In contrast to the various proposed techniques of anti-forensics of JPEG detection, few techniques to counter DJPEG detection have been proposed. Sutthiwan *et al.* [36] used a very simple shrink and zoom method to deceive the detector, but it displayed low undetectability. Li *et al.* [37] only targeted the DJPEG with the same quantization matrix. Lastly, Fan *et al.* [10] mentioned that their proposed anti-forensics of JPEG detection was also appropriate in the anti-forensics of feature-based DJPEG detection. Thus, the anti-forensics of a CNN-based DJPEG detection technology must be developed.

C. JPEG RESTORATION

JPEG restoration tasks focus on visual quality, especially in the low-QF JPEGs. CNNs for JPEG restoration using a feed-forward CNN and low-level features were proposed in [11]–[13]. Adding the DCT domain to the CNN branch improved the restored visual quality [14], [38].

Moreover, the frequency distributions of local patches were estimated by means of cross-entropy learning and were used in the encoder-decoder network for restoration [39]. In addition, GAN networks were proposed [40], and several networks that target different JPEG QFs were combined for generalization [41]. They usually used mean squared error (MSE) loss for the reconstruction, which is appropriate in improving visual quality.

III. PROPOSED METHOD

A. FRAMEWORK

As shown in Fig. 2, our proposed method consists of two parts: EDSR [16] based images reconstruction, DCT constraints. Let's designate I, I^{uncmp}, and $\hat{I}$ as the input JPEG image, the corresponding uncompressed image, and its reconstructed image in the proposed method, respectively. I is decompressed so that it is not truncated and rounded to give the network more abundant information. We first adopt the simplified EDSR network, represented as EDSR_{sub}, to reconstruct I into $\hat{I}^{recon}$ as follows:

$$\hat{I}^{recon} = \text{EDSR}_{sub}(I). \tag{1}$$

Next, the DCT coefficients, $\hat{I}^{recon}$, are modified to constrain the range of coefficients, and then we have final reconstructed image $\hat{I}$ as follows:

$$\hat{I} = \text{IDCT}(\text{DCT}_{const}(\text{DCT}(\hat{I}^{recon}))), \tag{2}$$

where DCT and IDCT represent the DCT operation and the inverse of the DCT operation. The structure and role of each component is specified in Sec. II-C and III-C.

The network is trained with the loss function L_{total}. The loss function L_{total} is composed of three different functions, and this is how we represent the three types of losses: the reconstruction loss is L_{recon}, the histogram loss is L_{hist}, and the deblocking loss is L_{deblk}. L_{recon} is the MSE loss function, which minimizes the pixel difference between $\hat{I}^{recon}$ and I^{uncmp} for visual quality. L_{hist} is designed to minimize the statistical difference between $\text{DCT}(\hat{I}^{recon})$ and $\text{DCT}(I^{uncmp})$. Lastly, L_{deblk} is devised to erase the JPEG blocking artifacts remaining in the borders of 8×8 JPEG blocks. The calculation of L_{hist} and L_{deblk} are specified in Sec. III-D and III-E. The goal of the training is to minimize L_{total}, which is defined on the basis of the hyper-parameters as follows:

$$L_{total} = \lambda_{recon} L_{recon} + \lambda_{hist} L_{hist} + \lambda_{deblk} L_{deblk}, \tag{3}$$

where $\lambda_{recon} = 1.0$, $\lambda_{hist} = 1.0$, and $\lambda_{deblk} = 0.06$. Each of the hyper-parameters are selected by several experiments

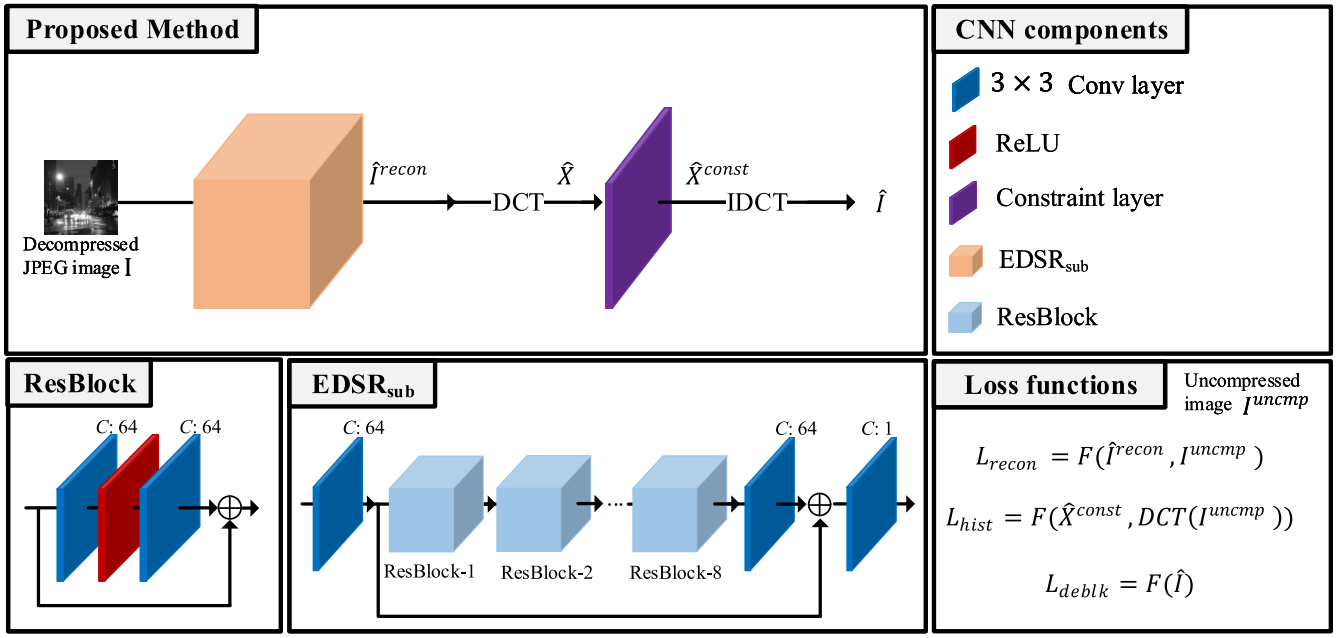

FIGURE 2. An overview of our proposed method network and proposed loss functions. The main structure of the proposed network, which is demarcated by its apricot color, is the EDSR network. Additionally, DCT constraints are applied after the EDSR network. The final output of the network is $\hat{I}$, and it is used in the calculation of the loss function. Three loss functions are used in training, namely L_{recon}, L_{hist}, and L_{deblk}. L_{recon} and L_{hist}, which were calculated with its uncompressed label data; L_{deblk} was calculated by the output itself.

and chose hyper-parameters that have high undetectability and less visual quality degradation which the detail is in the Sec. IV-E.

B. EDSR-BASED IMAGE RECONSTRUCTION

JPEG decoders, such as those in OpenCV and PIL, decompress images with 8-bit precision (0-255). To accomplish this, pixels are truncated and rounded after IDCT operation. We would like to note that there are useful traces in removed pixels that can be used by anti-forensics methods. So, we decompress JPEG images with 32-bit precision and normalize them by dividing them by 255. We feed these decompressed images to provide the network with more comprehensive information.

The main stream of the proposed network is EDSR$_{sub}$, which is inspired by the EDSR [16] for the task of producing super-resolution images. In this paper, we choose the EDSR network because it performs well in super-resolution though it has a simple structure. We also focused on the proposed loss functions' effect on undetectability because it has not been analyzed. To maintain the image resolution, we remove the up-sampling layer of the EDSR. As the network is fully convolutional and maintains resolution throughout its entirety, we could apply JPEG image of any resolution. In constructing the specific structure of the network, we use a 3 × 3 kernel size, 64 channels in all convolution layers, and eight residual blocks. As mentioned in various papers on super resolution and JPEG artifact removal [12], [14], [16], the use of batch normalization [42] is not apposite in addressing image restoration problems as it can lead to lower visual quality, so we remove batch normalization from our framework.

Reconstruction loss, $L_{recon} = \text{MSE}(\hat{I}^{recon}, I^{uncmp})$, is defined to maintain the visual quality of I^{uncmp}.

C. DCT CONSTRAINTS

After EDSR$_{sub}$, we transform $\hat{I}^{recon}$ into the DCT domain. Then, we get $\hat{X}$, the DCT coefficient of $\hat{I}^{recon}$, and clamp $\hat{X}$ to be in the range obtained from the rounding error in which the uncompressed DCT coefficients can exist. The rounding error range is $(-0.5, 0.5)$, and the uncompressed coefficients must be within the following range:

$$X_{i,j} - \frac{1}{2}Q_{i,j} \le X_{i,j}^{uncmp} \le X_{i,j} + \frac{1}{2}Q_{i,j}, \tag{4}$$

where $X_{i,j}$, $X_{i,j}^{uncmp}$, and $Q_{i,j}$ are the (i,j)th DCT subband coefficients of the JPEG image, I, the uncompressed image, I^{uncmp}, and the quantization matrix of I, respectively. We set the range boundaries as $low_{i,j} = X_{i,j} - \frac{1}{2}Q_{i,j}$ and $high_{i,j} = X_{i,j} + \frac{1}{2}Q_{i,j}$. This constraint is widely used in JPEG anti-forensics and restoration tasks for improving the performance of visual quality [10], [14], [38]. Therefore, we include this constraint and transform the output DCT coefficients such that they fall into the constraint range. On the other hand, we find that a restrictive constraint, as in the above equation, detracts from the level of undetectability, as shown in Sec. IV-E. Therefore, we add the trainable parameter α for soft clamping as follows:

$$\hat{X}_{i,j}^{const} = \begin{cases} (1-\alpha)low_{i,j} + \alpha\hat{X}_{i,j}, & \hat{X}_{i,j} < low_{i,j} \\ \hat{X}_{i,j}, & otherwise \\ (1-\alpha)high_{i,j} + \alpha\hat{X}_{i,j}, & \hat{X}_{i,j} > high_{i,j}, \end{cases} \tag{5}$$

where α is a scaling parameter that is initialized with 0.1, as in [14]. Finally, IDCT is performed on the output of the constraints to get the final image, $\hat{I}$.

D. HISTOGRAM SMOOTHING

Since DJPEG detection mostly uses the DCT histogram as a feature, we focused on the learning of the DCT histogram distribution of uncompressed images to deceive the detectors. In Sec. II-A, the anti-forensics of JPEG detection typically uses dithering noise in the DCT coefficients in order to resemble the uncompressed image of DCT distribution that approximates the Laplacian distribution [8]. Our approach is to try to match the DCT distribution with supervised learning.

To learn the histogram distribution, it is beneficial to calculate the histograms with the differentiable layer. The DCT histogram can be approximated by the CNN, as mentioned in [33], [34]. Specifically, we collect the same frequency bins of the DCT domain into the same axis, which results in the $(N_W = W/8, N_H = H/8, 64)$ size, and we denote the collected c_{th} channel DCT bin as D_c. After collecting each frequency value, we extract the cumulative histogram bins through the following equation:

$$S_{c,b} = sigmoid(\gamma * (D_c - b)), \tag{6}$$

where c is the channel number with the range $(1, 64)$, b is the histogram bin values with the range $(-60, 60)$, and γ is the parameter of the sigmoid that makes the sigmoid function discrete, which approximates 0 (if $D_c - b$ is negative) or 1 (if $D_c - b$ is positive). In previous papers, the γ value was large enough (10^6) to be discrete. However, in our approach, we apply it to the end of the network, and this could harm the training because of the large gradient. Therefore, we set a slightly lower value, namely, $\gamma = 10^2$.

Next, we average the calculated sigmoid values of $S_{c,b}$ to obtain the cumulative DCT histogram bins. Then, the difference of the sequential bins converts the cumulative bins to ordinary ones and is normalized as exhibited below.

$$H^{cm}_{c,b} = \frac{1}{N_W * N_H} \sum_{i=1}^{N_W} \sum_{j=1}^{N_H} S_{c,b}(i,j), \tag{7}$$

$$H = \{h \mid h_{c,b} = \frac{1}{64}(H^{cm}_{c,b+1} - H^{cm}_{c,b}),\ \forall c, b\}, \tag{8}$$

where H^{cm} is the approximated cumulative DCT histogram and H is the final approximated normalized DCT histogram of dimension [64, 120].

We extract the DCT histogram of the network output and the histogram of the corresponding uncompressed input image. We do not weight a specific histogram bin; rather, we use the L1 loss function for back-propagation in the learning of the histogram distribution as displayed below:

$$L_{smth} = L1(\hat{H}, H^{target}). \tag{9}$$

The histogram distribution learning with the L1 loss may harm the image quality because there is no spatial information. However, the constraint on the DCT coefficients limits the effect of the variation, and the histogram distribution could be learned with less of a decline in visual quality.

Additionally, as mentioned in Sec. II-A, we also add the calibration feature in histogram loss [20]. The lower right area of the 8×8 blocks in the DCT bins indicates the high frequency region, and we represent the segmented high-frequency histograms as H_{high}, and their size is $(28, 120)$. We calculate the feature using the high-frequency histograms of the output and its calibrated version, which is cropped by 4 pixels horizontally and vertically as follows:

$$L_{cal} = \frac{1}{28} \sum_{k \in high} \left| var(\hat{H}_{high,k}) - var(\hat{H}^{cal}_{high,k})) \right|. \tag{10}$$

To summarize this section, the histogram loss function is defined as below:

$$L_{hist} = s_{mse}\lambda_{smth}L_{smth} + \lambda_{cal}L_{cal}, \tag{11}$$

where $\lambda_{smth} = 1.0$ and $\lambda_{cal} = 0.0003$. For a higher undetectability in low-QF JPEGs and a higher visual quality in high-QF JPEGs, we scale the smoothing loss functions by MSE of the input pixel and the target, which is $s_{mse} = \text{MSE}(I, I^{uncmp}) \times 2,500$. We do not scale the calibration loss because comparable effect on loss was found in all QFs.

E. JPEG DEBLOCKING

We found that smoothing the DCT histogram does not sufficiently deceive the DJPEG detectors especially in low-QF JPEGs, which is demonstrated in Sec. IV-E. To achieve a higher level of undetectability, we add a TV minimization-based JPEG deblocking loss. TV minimization is applied for smoothing images [43] and JPEG artifact removal [10], [44]. To reduce the amount of JPEG artifacts, we minimize the total variation of the JPEG image particularly in the block boundary, which, unlike uncompressed images, has a large pixel difference. We calculate the total variation of each pixel by the following equation:

$$\begin{aligned} v_{i,j} = &(I_{i-1,j} + I_{i+1,j} - 2I_{i,j})^2 \\ &+ (I_{i,j-1} + I_{i,j+1} - 2I_{i,j})^2. \end{aligned} \tag{12}$$

where $I_{i,j}$ is the (i,j)th pixel value of the image. In the cases that escape from the image such as $i = 0$, we pad the image with the sequential pixel value. We use squared rather than absolute differences as a total variation value to weigh the gradient in the large difference region, which is the boundary part of the block. Moreover, we average the total variation of images and use it as a loss of JPEG deblocking as below:

$$L_{deblk} = s_{mse} \times \text{Mean}(\hat{v}_{i,j}). \tag{13}$$

IV. EXPERIMENTS

To demonstrate the effectiveness of the proposed method, we compare it with DDCN for the JPEG restoration task [14] and the subgradient anti-forensics method of Fan *et al.* [10]. JPEG restoration generates high-quality images, but it is rarely discussed whether it is able to remove block artifacts.

Therefore, we evaluate their undetectability to compare it to ours. Fan *et al.* [10] achieved the highest undetectability among all previous studies of anti-forensics of JPEG detection and used for the main comparison. For clarification, we represent them as AF_{our}, R_{ddcn}, and AF_{fan}, respectively.

A. SETTINGS

1) DATASETS

We use the BossBase 1.01 [17] and BOWS2 [18] datasets for the experiments. Each dataset contains 10,000 uncompressed gray scale and 512×512-sized images. We split the dataset into 16,000, 1,000, and 3,000 images as the training, validation, and test sets. In addition, we crop each of the images into quarters, which results in four 256×256 images. Therefore, the total training, validation, test dataset consists of 64,000, 4,000, and 12,000 images. We then transform each uncompressed image into JPEG and DJPEG format by using the PIL library. We use {50, 60, 70, 80, 90} as the first JPEG QF and {75, 85, 95} as the second QF to avoid the duplicated quantization matrix, which is a completely different problem.

When recompressing JPEGs, the alignment of the compressed blocks could be the same or could vary from that of previous blocks. If the new alignment is different, the block artifacts will still establish another alignment: they will form a different feature with the one they align with and show a different aspect of detection. For convenience, the aligned DJPEG is represented as DJPEG and the non-aligned one is represented as non-aligned DJPEG.

For the non-aligned DJPEG, we randomly select a pair of indexes, $(a, b) \in \{\{x \in Z | 0 \leq x \leq 7\}^2 - (0, 0)\}$, and crop the JPEG image from (a,b) position to (256-(8-a), 256-(8-b)) with a 248 width and height. After cropping, we recompressed the cropped JPEG with the quality factors, {75, 85, 95}. To make the same resolution of JPEG and non-aligned DJPEG in the non-aligned task, the JPEG is cropped from positions (0,0) or (8,8) to positions (248,248) or (256,256).

2) EVALUATION METRICS

We consider two types of evaluation metrics, undetectability in JPEG and DJPEG detectors and the visual quality of their respective images. The output of the models is saved as uncompressed format, a PGM file, and is tested with the JPEG detectors. In the case of DJPEG undetectability, the reconstructed image is recompressed with three $QF \in \{75, 85, 95\}$ and tested with the DJPEG detectors. The non-aligned version of the reconstructed image is constructed by the same method used in creating a non-aligned DJPEG. We calculate PSNR and SSIM for evaluating the visual quality of the reconstructed uncompressed images.

3) IMPLEMENTATION DETAILS

The proposed method is trained with a batch size 16 up to 10 epochs. It is optimized by Adam with an initial learning rate of 10^{-4} and default hyper-parameters. The proposed method has no prior knowledge of target detectors for generalization. Therefore, the trained model is selected for testing when validation loss of histogram and deblocking minimized. Our proposed method is implemented on the Pytorch framework on GTX 1080 Ti GPU.

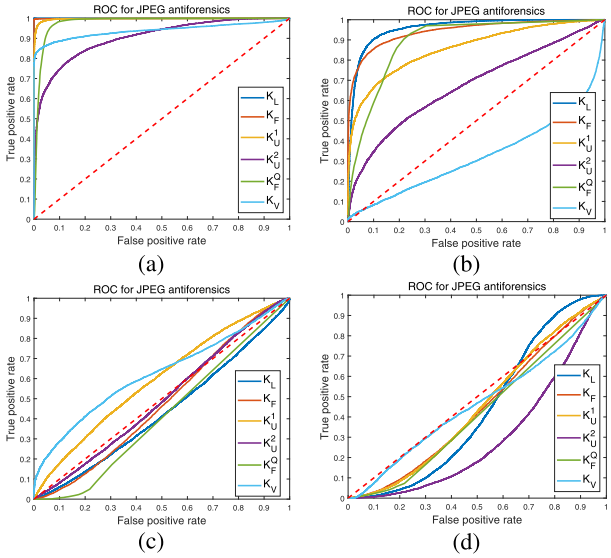

FIGURE 3. The ROC curves of JPEG detectors with QF 60 JPEG images and their reconstructed images. It is drawn based on the thresholding of feature values extracted from the uncompressed and reconstructed images. (a) is JPEG, (b) is R_{ddcn} [14], (c) is AF_{fan} [10], and (d) is AF_{our}. The dotted line represents the random guess, which is the optimal result. The JPEG and R_{ddcn} can be distinguished by thresholding, but AF_{fan} and AF_{our} is almost similar with random guessing.

B. ANTI-FORENSICS OF JPEG AND VISUAL QUALITY

For the evaluation of the undetectability of JPEG compression, we use the six JPEG detectors mentioned in Sec. II-A as follows:

- K_L: Calibration feature based detector [20],
- K_F: Block artifact measure [5],
- K_U^1, K_U^2: Gradient feature-based detector [23],
- K_F^Q: Quantization matrix estimation [5],
- K_V: Total variation of recompressed image-based detector [22].

In Table 1, we report the minimum decision error rate of the six JPEG detectors for the JPEG images, R_{ddcn}, AF_{fan}, and AF_{our}. The detectors classify based on the thresholding of each feature value. Therefore, we compute features of 12,000 anti-forensically processed test images and their corresponding 12,000 uncompressed images. Then, we find the threshold and minimum decision error by drawing ROC curves, as shown in Fig. 3, which are graphs of the six JPEG detectors with images of QF 60. The detectors that approximate a minimum error rates of 0.5 have difficulties in judging uncompressed and JPEG images. We also specify the PSNR and SSIM for each case. Additionally, we exhibit an example of reconstructed results and (2,2) subband DCT histograms in Fig. 4.

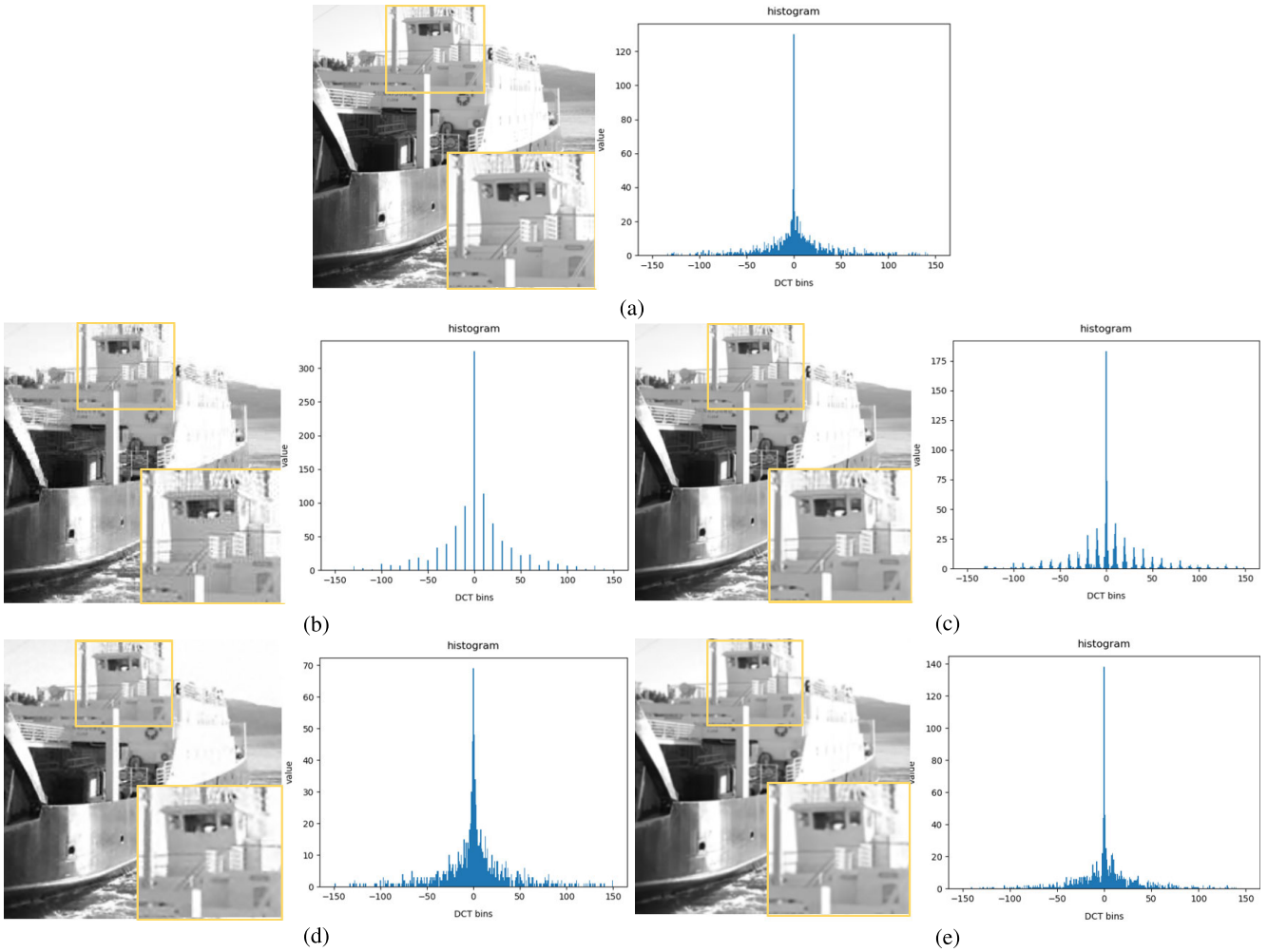

FIGURE 4. The example of QF 60 JPEG and reconstructed images with their DCT histogram of a (2,2) subband. (a) is an uncompressed image. The image quality is high, and the DCT histogram is a continual function that resembles the laplacian distribution. (b) is the JPEG, and the DCT histogram is discrete. (c) is reconstructed by R_{ddcn} [14]; its visual quality has been increased, but the DCT histogram still has discrete features. (d) is reconstructed by AF_{fan} [10]; its visual quality is lower than the JPEG, but the DCT histogram is continual and resembles the laplacian distribution except it shows slight difference to the uncompressed images. (e) is reconstructed by AF_{our}, the image has fewer noises, and the DCT histogram is much more similar to uncompressed images than other algorithms are.

First, we note that all the detectors easily classify JPEG images as JPEG with a low minimum decision error and high area under the curve (AUC) in the ROC curve. R_{ddcn} restores the damaged images successfully, as is shown by the improvements in PSNR and SSIM, and the example image results in Fig. 4. Nevertheless, it fails to deceive the JPEG detectors except for K_U^2 and K_V. Moreover, in Fig. 4, we could find that reconstructed images using JPEG restoration still have a discrete feature similar to JPEGs. Therefore, JPEG restoration better resembles the uncompressed one in the pixel domain than that of the DCT domain, especially the case in the high QF JPEG, and the degree of variation in the DCT histogram is less than pixel domain. As a result, it is detected effectively by the DCT domain detectors, which are K_L and K_F^Q.

By contrast, AF_{fan} had aspects that were similar to random guessing and achieved similar feature values with the uncompressed images in all of six features [10], and the performance AF_{our} followed suit. On average, AF_{fan} performed slightly better than AF_{our}. This is because AF_{fan} showed a higher minimum decision error in both K_L and K_F, as they employed prior knowledge of K_L and K_F as a feature distribution of uncompressed image for thresholding in the subgradient method. Furthermore, the K_U^2 feature magnitude distribution of AF_{our} was slightly lower than in uncompressed ones. In terms of the visual quality, AF_{fan} degraded it more than the JPEG images, especially in the case of high-QF JPEGs. However, our approach allowed us to reconstruct images with a decreased loss in visual quality, as shown in Table 1.

On the one hand, as shown in Fig. 4, AF_{our} produced slightly blurry images in comparison to other methods. However, images reconstructed by AF_{fan} have noisy signals, but AF_{our} could reconstruct smooth images that obviate

TABLE 1. The minimum decision error rate of six JPEG detectors for JPEG images, reconstructed images of R_{ddcn} [14], AF_{fan} [10], and AF_{our}, respectively. Values closer to 0.5 are better for anti-forensics methods.

QF	Method	JPEG detector							Visual Quality	
		K_L [20]	K_F [5]	K_U^1 [23]	K_U^2 [23]	K_F^Q [5]	K_V [22]	Mean	PSNR	SSIM
50	JPEG	0.0002	0.0034	0.0202	0.1589	0.0584	0.1062	0.0579	36.53	0.9378
	R_{ddcn}	0.1316	0.2352	0.2400	0.3676	0.1598	0.3816	0.2526	**38.28**	**0.9527**
	AF_{fan}	**0.4552**	**0.4802**	**0.4461**	**0.4871**	0.4156	0.3950	**0.4465**	35.79	0.9256
	AF_{our}	0.4012	0.3909	0.4238	0.3300	**0.4263**	**0.4573**	0.4049	36.52	0.9366
60	JPEG	0.0008	0.0038	0.0230	0.1735	0.0542	0.0880	0.0572	37.30	0.9466
	R_{ddcn}	0.0984	0.1220	0.2152	0.3595	0.1563	0.3430	0.2157	**39.05**	**0.9594**
	AF_{fan}	**0.4537**	**0.4609**	0.4344	**0.4841**	0.4157	0.3930	**0.4403**	36.45	0.9339
	AF_{our}	0.3974	0.4393	**0.4358**	0.3369	**0.4271**	**0.4605**	0.4162	37.02	0.9437
70	JPEG	0.0032	0.0045	0.0344	0.1963	0.0486	0.1060	0.0655	38.37	0.9564
	R_{ddcn}	0.1233	0.1631	0.2310	0.3769	0.1434	0.3020	0.2233	**40.11**	**0.9673**
	AF_{fan}	**0.4540**	**0.4651**	0.4296	**0.4848**	0.4176	0.3750	**0.4377**	37.32	0.9438
	AF_{our}	0.4182	0.4312	**0.4382**	0.3611	**0.4447**	**0.4732**	0.4278	38.00	0.9552
80	JPEG	0.0151	0.0114	0.0480	0.2377	0.0409	0.0700	0.0705	39.95	0.9677
	R_{ddcn}	0.1599	0.1789	0.2650	0.4108	0.1068	0.3385	0.2433	**41.66**	**0.9758**
	AF_{fan}	**0.4557**	**0.4737**	0.4347	**0.4865**	0.4201	0.3580	**0.4381**	38.50	0.9548
	AF_{our}	0.3979	0.3935	**0.4618**	0.3930	**0.4220**	**0.3990**	0.4112	39.35	0.9651
90	JPEG	0.0937	0.0494	0.1124	0.3517	0.0315	0.0660	0.1175	43.03	0.9821
	R_{ddcn}	0.2137	0.1448	0.2760	0.4521	0.0605	0.2250	0.2287	**44.52**	**0.9864**
	AF_{fan}	0.4104	**0.4600**	0.4090	**0.4754**	0.4169	0.3580	0.4216	40.58	0.9694
	AF_{our}	**0.4268**	0.4221	**0.4501**	0.4375	**0.4513**	**0.4205**	**0.4347**	42.27	0.9813

noisy signals. Additionally, in the example results shown in Fig. 4, the AF_{fan} could make DCT histogram similar to an anonymously uncompressed distribution but has distance with DCT histogram of its uncompressed version. In contrast to the AF_{fan}, AF_{our} could reconstruct a more similar DCT histogram of its uncompressed one and improve visual quality.

C. ANTI-FORENSICS OF DJPEG

We consider a DJPEG detector and a steganalysis detector to evaluate undetectability in both the DCT and pixel domains as follows:

- K_P: DJPEG detector with DCT histogram [34],
- K_S: Steganalysis with pixel domain images [45].

DJPEG detection in the pixel domain is more difficult than it is in the DCT domain because the former has fewer features [33]. Therefore, we employed SRNet [45], which is the state-of-the-art network of steganalysis, as it is highly effective at capturing residual noise and is suitable for DJPEG detection. According to [34], a mixed QF JPEG dataset is possible for training. Therefore, we mixed all of the QF JPEG {50, 60, 70, 80, 90, 75, 85, 95} and DJPEG {50, 60, 70, 80, 90} recompressed with {75, 85, 95} as mentioned in Sec. IV-A1 for training both networks. The networks were trained with binary cross-entropy loss, and the outputs with the Softmax represent the probabilities of JPEG and DJPEG.

In Tables 2 and 3, we report the accuracy rates and minimum decision error rate of two detectors and for the AF_{our}, R_{ddcn}, and AF_{fan} methods; the accuracy rate denotes the rate at which the detectors classify the manipulated images as DJPEG and the minimum decision error rate is calculated similarly as in the previous section using the probability values of JPEG and the other methods. Although K_P exhibited a better performance than K_S in the case of low-QF JPEGs, both models successfully found the trace of DJPEG compression.

R_{ddcn} could not deceive the detectors in the low {75, 85} second QF DJPEG images in either detector as is indicated by the fact that the example result in Fig. 4 still resembles JPEG DCT histogram. However, for the high second QF 95, they were able to deceive detectors to some extent. As JPEG restoration had less variation in the DCT domain, they could deceive K_S better than K_P when recompressed with high QF.

The anti-forensics of JPEG detection exhibited better undetectability than the JPEG restoration task in almost all cases. Furthermore, it was highly effective at deceiving the K_P. On the other hand, for K_S, they showed reasonable results but had a higher detection rate than K_P, as dithering noise in the DCT histogram resembles the natural DCT histogram; however, it seemed to harm the pixel domain. Also, they showed less undetectability in both domains when compressed with low-QF.

In the proposed method, the undetectability in the K_P proved to be similar with that of AF_{fan} except for the QF2 95 minimum decision error rate, as they both effectively deceived the detectors. In contrast, for K_S, our method has shown far higher undetectability in almost all QF cases. AF_{fan} displayed poor undetectability in K_S in the low-QF JPEG, but our method solved the problem with higher visual quality. When reconstructed images were recompressed with the high QF (95), most of the images were classified as JPEG but showed a low minimum decision error rate that has a little distance with the ground truth JPEG distribution.

D. ANTI-FORENSICS OF NON-ALGINED DJPEG

We consider the same detectors as the those listed in the previous section, namely, K_P and K_S. However, in this section, they

TABLE 2. The detection accuracy rate and the minimum decison error rate of DJPEG detectors for DJPEG images and reconstructed images of R_{ddcn} [14], AF_{fan} [10], and AF_{our}, respectively. The left value is the accuracy rate and the right is the minimum decision error rate. The reconstructed and normal JPEG images with QF1 are recompressed by QF2. Smaller values for accuracy rate and close to 0.5 for minimum decision error rate have higher undetectability to the detectors.

JPEG quality		Double JPEG detector							
QF2	QF1	K_P [34]				K_S [45]			
		DJPEG	R_{ddcn}	AF_{fan}	AF_{our}	DJPEG	R_{ddcn}	AF_{fan}	AF_{our}
75	50	0.999/0.003	0.983/0.015	0.359/0.194	**0.073/0.349**	0.992/0.026	0.970/0.034	0.768/0.132	**0.157/0.409**
	60	0.999/0.002	0.975/0.019	**0.061**/0.354	0.080/**0.381**	0.999/0.010	0.950/0.045	0.548/0.228	**0.032/0.457**
	70	0.994/0.009	0.588/0.119	**0.011**/0.463	0.021/**0.471**	0.992/0.017	0.712/0.141	0.354/0.331	**0.059/0.444**
	80	0.995/0.008	0.760/0.071	**0.016/0.456**	0.027/0.439	0.993/0.017	0.700/0.160	0.243/**0.392**	**0.204**/0.379
	90	0.987/0.013	0.962/0.024	**0.022/0.441**	0.041/0.426	0.973/0.034	0.850/0.090	0.224/0.360	**0.038/0.494**
85	50	1.00/0.00	0.536/0.045	**0.00**/0.491	0.001/**0.492**	0.998/0.008	0.688/0.138	0.334/0.275	**0.140/0.353**
	60	1.00/0.00	0.632/0.049	**0.00/0.490**	0.001/0.399	0.999/0.007	0.686/0.130	0.271/0.317	**0.062/0.433**
	70	1.00/0.00	0.929/0.015	**0.00/0.492**	0.00/0.454	0.987/0.013	0.800/0.079	0.237/0.365	**0.012/0.418**
	80	0.999/0.001	0.584/0.046	**0.00**/0.488	0.001/**0.492**	0.997/0.008	0.582/0.158	0.122/0.434	**0.024/0.460**
	90	0.997/0.001	0.854/0.028	**0.00**/0.489	0.001/**0.491**	0.997/0.006	0.665/0.137	**0.031/0.490**	0.150/0.314
95	50	1.00/0.00	0.094/0.051	**0.00/0.282**	0.001/0.279	0.977/0.001	0.310/0.186	0.179/0.250	**0.093/0.262**
	60	1.00/0.00	0.095/0.040	**0.00/0.317**	0.001/0.136	0.976/0.002	0.238/0.171	0.194/**0.267**	**0.088**/0.225
	70	1.00/0.00	0.190/0.028	**0.00/0.337**	0.001/0.136	0.959/0.003	0.095/0.175	0.186/0.285	**0.035/0.305**
	80	1.00/0.00	0.328/0.019	**0.00/0.335**	0.001/0.115	0.926/0.005	0.022/0.201	0.048/**0.350**	**0.009**/0.331
	90	1.00/0.00	0.829/0.002	**0.00/0.427**	0.001/0.201	0.982/0.003	0.046/0.286	0.006/**0.420**	**0.002**/0.400

TABLE 3. The detection accuracy rate of two detectors for JPEG.

JPEG QF	Detector	
	K_P [34]	K_S [45]
50	1	0.8820
60	1	0.9172
70	1	0.9577
80	0.9997	0.9927
90	0.9996	0.9841
75	0.9863	0.9551
85	0.9992	0.9851
95	0.9995	0.9995

were trained with a mixed QF non-aligned DJPEG dataset and mixed QF cropped JPEG dataset, similar to those observed in aligned DJPEG. In addition, the networks were trained with binary cross-entropy loss in the same way as the aligned ones, and the evaluation metrics and processes are also the same. We represent the non-aligned DJPEG detectors as follows:

- K_P^{na}: non-aligned DJPEG detector with DCT histogram [34],
- K_S^{na}: non-aligned DJPEG with pixel domain images [45].

In Table 4 and 5, we report the accuracy rates and the minimum decision error rate of two detectors and for the AF_{our}, R_{ddcn}, and AF_{fan} methods. As mentioned in the previous section, DJPEG detection in the pixel domain is more difficult than the DCT domain, but in the non-aligned cases, the pixel domain is much easier [33] because of the multiple block artifacts. As a result, the K_S^{na} could classify more effectively than the K_P^{na}, and K_S^{na} could classify in almost all cases. Moreover, the K_P^{na} could not identify JPEG and non-aligned DJPEG images in the low second QF {75, 85} as the detection accuracy rate and the minimum decision error rate of DJPEG is high and the cases are not amenable to determining undetectability. Therefore, for K_P^{na}, we mainly focused on the second QF 95 cases.

In contrast to DJPEG detectors, R_{ddcn} was able to deceive non-aligned DJPEG detectors. The R_{ddcn} showed a reasonable level of undetectability in K_S^{na} for all of the QF cases but showed less undetectability in the K_P^{na} method, as they possesses less variation in terms of the DCT domain. However, AF_{fan} performed more poorly than R_{ddcn} in the K_S^{na} but similarly in the K_P^{na}. It displayed high detectability, especially in the low second QF {75, 85}, and is not appropriate for non-aligned DJPEG anti-forensics.

The proposed method showed the highest level of undetectability in both detectors. To be specific, it showed less detection accuracy and high minimum decision error than the R_{ddcn} in almost all cases except low second QF {75, 85} in K_P^{na}, but they are not suitable for measuring undetectability as mentioned. Besides, it showed higher undetectability than the AF_{fan} in all of the QF cases. To summarize this section, we demonstrated that the loss functions for visual quality, which is MSE, is suitable for deceiving non-aligned DJPEG detectors, and the AF_{fan} algorithm is not appropriate.

E. ABLATION STUDY

We explored several loss functions for designing the training methodology for our task of finding the optimal undetectability in JPEG and DJPEG detectors. For non-aligned DJPEG, the visual quality loss function was adequate for deceiving detectors, and we skipped this exploration. In each trial, we used images with JPEG QF 60 for reconstruction and recompressed with QF2 75, which resulted in a low accuracy rate in the evaluated methods.

The results of the effect of each loss functions is shown in Table 6, where L_{recon}, $L_{recon} \cup L_{hist}$, and $L_{recon} \cup L_{hist}$ indicate the use of only the noted loss functions with same

TABLE 4. The detection accuracy rate and the minimum decision error rate of non-aligned DJPEG detectors for DJPEG images and reconstructed images of R_{ddcn} [14], AF_{fan} [10], and AF_{our}, respectively. The left value is the accuracy rate and the right is the minimum decision error rate. The reconstructed and normal JPEG images with QF1 are recompressed by QF2. Smaller values for accuracy rate and close to 0.5 for minimum decision error rate have higher undetectability to the detectors.

JPEG quality		Double JPEG detector							
QF2	QF1	K_P^{na} [34]				K_S^{na} [45]			
		DJPEG	R_{ddcn}	AF_{fan}	AF_{our}	DJPEG	R_{ddcn}	AF_{fan}	AF_{our}
75	50	0.993/0.076	**0.665/0.376**	0.940/0.205	0.720/0.376	0.998/0.010	0.199/0.422	0.828/0.117	**0.123/0.469**
	60	0.981/0.124	**0.627/0.415**	0.903/0.243	0.751/0.358	0.998/0.010	**0.143/0.453**	0.758/0.143	0.144/0.427
	70	0.931/0.221	**0.583/0.444**	0.829/0.309	0.717/0.379	0.996/0.017	0.155/0.447	0.604/0.201	**0.104/0.462**
	80	0.857/0.292	**0.533/0.474**	0.709/0.384	0.697/0.393	0.985/0.031	**0.125/0.463**	0.400/0.289	0.129/0.436
	90	0.653/0.417	**0.498/0.494**	0.626/0.427	0.554/0.465	0.805/0.126	0.127/0.460	0.210/0.399	**0.080/0.491**
85	50	0.999/0.017	0.624/0.298	0.889/0.158	**0.517/0.342**	1.00/0.002	0.079/0.422	0.537/0.113	**0.046/0.473**
	60	0.998/0.022	0.623/0.299	0.847/0.184	**0.510/0.344**	1.00/0.002	0.063/0.441	0.472/0.133	**0.032/0.462**
	70	0.991/0.051	0.585/0.320	0.736/0.243	**0.454/0.366**	0.999/0.003	0.073/0.427	0.333/0.193	**0.034/0.473**
	80	0.890/0.158	**0.373/0.417**	0.569/0.321	0.454/0.368	0.997/0.005	0.056/0.444	0.172/0.282	**0.025/0.461**
	90	0.652/0.283	**0.300/0.460**	0.425/0.390	0.328/0.430	0.929/0.028	0.053/0.441	0.058/0.384	**0.014/0.475**
95	50	1.00/0.002	0.422/0.134	0.392/0.160	**0.208/0.279**	1.00/0.001	0.036/0.336	0.280/0.067	**0.003/0.429**
	60	1.00/0.002	0.456/0.126	0.365/0.167	**0.172/0.282**	1.00/0.001	0.033/0.356	0.261/0.076	**0.002/0.466**
	70	0.998/0.004	0.513/0.118	0.299/0.188	**0.176/0.264**	1.00/0.001	0.050/0.340	0.221/0.108	**0.001/0.455**
	80	0.996/0.006	0.477/0.122	0.190/0.237	**0.097/0.314**	1.00/0.001	0.042/0.387	0.151/0.180	**0.00/0.489**
	90	0.925/0.025	0.328/0.187	0.115/0.319	**0.056/0.353**	0.980/0.004	0.058/0.350	0.081/0.257	**0.002/0.473**

TABLE 5. The detection accuracy rate of two non-aligned detectors for JPEG.

JPEG QF	Detector	
	K_P^{na} [34]	K_S^{na} [45]
50	1	0.9992
60	1	0.9992
70	1	0.9937
80	1	0.9927
90	0.9997	0.9987
75	0.5226	0.9269
85	0.7884	0.9917
95	0.9936	0.9985

hyper-parameter as the proposed method, and AF_{our}^{strict} indicates the proposed method with strict constraints, as explained below paragraph. As L_{recon} exhibited a poor level of undetectability, we propose two loss functions, histogram, and deblocking loss, to improve anti-forensics performance. Both functions, histogram and deblocking loss, increased the undetectability but degraded the visual quality. In the case of the JPEG detectors, histogram loss significantly improved undetectability except for K_F but deblocking loss compensated for the low undetectability performance in K_F. For the DJPEG detectors, histogram loss helped to deceive both detectors and seemed to improve better in the pixel domain. However, it decreased the undetectability in the pixel domain when it got too small because it increased the difference of DCT bins between the neighbor blocks, which created other block artifacts in the pixel domain. Deblocking loss also improved undetectability in both the DCT domain and pixel domain because of the reason mentioned in Sec. III-E and the fact cited in [10] (i.e., the minimization of the total variation have the effect of smoothing the DCT histogram effect). However, over smoothing degrades the visual quality; the undetectability of the DCT histogram domain, especially in high-QF JPEGs; and the undetectability level in the pixel domain. Therefore, we controlled the rate of each loss to prevent deviation from the optimal convergence point, and the balancing out of all losses led to the best performance.

The strict constraint of the DCT domain, in which the alpha was 0 in Eq.(5) helped to increase visual quality, as mentioned in Sec. III-C, but it could not deceive JPEG detectors, K_F, K_F^Q, and K_V, and showed less undetectability in the pixel domain for DJPEG detectors. With the strict constraint, the DC component distribution, which is the highest frequency bin of DCT, was still discrete after training, which made K_F^Q detectable. Moreover, dithering DCT coefficients in the strict range could be detected by K_V, as recompressing with the same QF cancels the effect of tampering [22]. Therefore, we applied the soft constraint rather than the strict one, and this improved undetectability.

F. GENERALIZATION

JPEGs in the real world are compressed with diverse QFs. Each JPEG compressed with different QFs has different block artifacts and could produce different results than what we expect. Additionally, the resolution of the JPEGs in the real world is also diverse rather than not fixed to 256×256. This section will explore the robustness and generalizability of our model through considering the out-of-distribution hyper-parameters of the dataset.

At first, we studied the robustness of our proposed method in relation to diverse QFs. To achieve optimal visual quality and undetectability in our method, each QF has to be trained with different hyper-parameters and datasets. Therefore, our proposed method was trained with the fixed QFs separately and with the fixed QF in a set {50, 60, 70, 80, 90}. For testing the robustness of our model, we tested reconstruction

TABLE 6. The minimum decision error of JPEG detectors, the detection accuracy rate and minimum decision error rate of DJPEG detectors, and the visual quality according to loss combinations with QF 60 JPEGs and recompressed with QF2 75 for DJPEG detectors.

Method	JPEG detector						DJPEG detector		Visual Quality	
	K_L [20]	K_F [5]	K_U^1 [23]	K_U^2 [23]	K_F^Q [5]	K_V [22]	K_P [34]	K_S [45]	PSNR	SSIM
L_{recon}	0.1205	0.0302	0.1822	0.3633	0.1461	0.3458	0.964/0.024	0.999/0.006	39.07	0.9594
$L_{recon} \cup L_{hist}$	**0.4349**	0.2885	0.4058	0.4659	0.4075	**0.4815**	0.709/0.081	0.289/0.365	38.47	0.9529
$L_{recon} \cup L_{deblk}$	0.2435	0.4201	0.3755	0.4365	0.1633	0.2439	0.122/0.370	0.594/0.187	37.29	0.9477
AF_{our}^{strict}	0.3226	0.1791	**0.4508**	**0.4964**	0.1848	0.1665	**0.080**/0.377	0.456/0.257	37.5	0.9495
AF_{our}	0.3974	**0.4393**	0.4358	0.3369	**0.4271**	0.4605	0.080/**0.381**	**0.032/0.457**	37.02	0.9437

TABLE 7. The minimum decision error in JPEG detectors, and the visual quality of the generalized input cases, mixed QF, and large resolution. The mixed QF is a set of QFs close to 60, which ranges from 56-64 to test the robustness of the model in diverse QF. The large resolution is 512 × 512, and it is selected for testing the robustness of the resolution.

Dataset			JPEG detector						Visual Quality	
resolution	QF	Method	K_L [20]	K_F [5]	K_U^1 [23]	K_U^2 [23]	K_F^Q [5]	K_V [22]	PSNR	SSIM
256×256	60	JPEG	0.0008	0.0038	0.0230	0.1735	0.0542	0.0880	37.30	0.9466
		AF_{our}	0.3974	0.4393	0.4358	0.3369	0.4271	0.4605	37.02	0.9437
	QF_{mixed}^{60}	JPEG	0.0008	0.0038	0.0230	0.1729	0.0543	0.1163	37.26	0.9461
		AF_{our}	0.3976	0.4385	0.4355	0.3378	0.4290	0.4600	37.00	0.9434
512×512	60	JPEG	0.00	0.0003	0.0018	0.0870	0.0250	0.0498	36.71	0.9464
		AF_{our}	0.3907	0.4555	0.4610	0.3237	0.4308	0.4393	36.35	36.71

TABLE 8. The detection accuracy rate and the minimum decision error rate in DJPEG detectors of the generalized input cases, mixed QF, and large resolution. The mixed QF is a set of QFs close to 60, which ranges 56-64 to test the robustness of the model in diverse QF. The large resolution is 512 × 512 and is for testing the robustness of the resolution.

Dataset				DJPEG detector	
resolution	QF2	QF	Method	K_P [34]	K_S [45]
256×256	75	60	DJPEG	0.999/0.002	0.999/0.010
			AF_{our}	**0.080/0.381**	**0.032/0.457**
		QF_{mixed}^{60}	DJPEG	0.992/0.007	0.985/0.023
			AF_{our}	**0.102/0.352**	**0.045/0.471**
	85	60	DJPEG	1.00/0.00	1.00/0.007
			AF_{our}	**0.001/0.399**	**0.062/0.433**
		QF_{mixed}^{60}	DJPEG	1.00/0.00	0.964/0.025
			AF_{our}	**0.001/0.406**	**0.066/0.423**
	95	60	DJPEG	1.00/0.00	0.976/0.002
			AF_{our}	**0.001/0.136**	**0.088/0.225**
		QF_{mixed}^{60}	DJPEG	1.00/0.00	0.865/0.017
			AF_{our}	**0.001/0.147**	**0.093/0.226**
512×512	75	60	DJPEG	1.00/0.00	1.00/0.001
			AF_{our}	**0.013/0.400**	**0.006/0.451**
	85	60	DJPEG	1.00/0.00	1.00/0.00
			AF_{our}	**0.00/0.447**	**0.033/0.374**
	95	60	DJPEG	1.00/0.00	1.00/0.00
			AF_{our}	**0.00/0.145**	**0.046/0.122**

with diverse QFs, but in a fashion similar to training QF. We defined the QFs that were similar as QF_{mixed}^{t}, which ranges from the $[t-4, t+4]$ integer and the $t \in \{50, 60, 70, 80, 90\}$. The rounding of the QF_{mixed}^{t} converges to t and could achieve any QF by ensemble. For example, for the QF 60 model, the QF_{mixed}^{60} JPEGs, which QF ranges 56-64 were employed in our testing.

Although each QF JPEGs contains different artifacts, as shown in Table. 7 and 8, the QF_{mixed} JPEGs could achieve similar visual quality and undetectability results through training QF. Therefore, the model features a degree of robustness of the similar to that of QF JPEGs, and combinations of each model could achieve a good performance in any QF JPEGs. In the scenario in which the QF of a JPEG is unknown, we could add the QF predictor model in front of the reconstruction model and pass over to the similar QF model, as is comparable to what is done in [41].

For analysing the resolution robustness, we examined the larger, 512 × 512 JPEGs, which are the original BossBase 1.01 [17] and BOWS2 [18] datasets. The main structure of our proposed method is composed of the EDSR model, which is the fully convolutional network and is applicable to any image resolution, as is stated in Sec. II-C. Therefore, it was only necessary for us to reconstruct the 512 × 512 QF 60 JPEG images and evaluate its the performance. As the resolution increases, the detectability increases because of increasing abundance of information. On the other hand, as shown in Table. 7 and 8, our proposed method has showed a similar level of undetectability in the large resolution. Therefore, our proposed method is also appropriate to reconstruct image of any resolution with high undetectability.

V. CONCLUSION

In this work, we proposed a CNN for anti-forensics of JPEG and DJPEG detection. The network is composed of EDSR and DCT constraints. We found that training only with a loss function for visual quality, which is MSE loss was adequate for deceiving the non-aligned DJPEG detectors, but it was inadequate for anti-forensics of JPEG and DJPEG detection. Therefore, we proposed two anti-forensical loss functions, histogram loss and deblocking loss. The histogram loss function helped to learn uncompressed DCT histogram distribution and increased undetectability in both the pixel and DCT domains. The deblocking loss function also helped to increase undetectability in both pixel and DCT domains by reducing the distance between boundary and center distribution. In addition, to improve undetectability, soft constraints of DCT was necessary as strict constraints had some limitations regarding their undetectability when facing several detectors.

The previous JPEG reconstruction tasks were introduced into two streams: JPEG restoration and anti-forensics of JPEG detection. JPEG restoration tasks only focused on visual quality, and it used only visual quality loss function in training. They improved visual quality, but the JPEG DCT histogram feature persisted after reconstruction. Therefore, they could not deceive JPEG and DJPEG detectors. However, they could deceive non-aligned DJPEG detectors. Anti-forensics of JPEG could make similar DCT histograms with uncompressed ones, but they differed little from the original. Therefore, they degraded the visual quality but removed the JPEG artifacts, especially in the DCT domain, that can be disguised in uncompressed images. However, they showed low undetectability in the pixel domain, especially in the non-aligned cases. Our work represents CNN-based anti-forensics of JPEG and DJPEG that can achieve high undetectability in both the DCT and pixel domains with less degradation to visual quality.

For the generalization to deal with real cases, we evaluated our model's robustness in the diverse QFs and resolution. Our model was robust to the QFs that are similar with trained one and was also robust to the large resolution.

REFERENCES

[1] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 6, pp. 1841–1854, Dec. 2012.

[2] Y. Liu, X. Zhu, X. Zhao, and Y. Cao, "Adversarial learning for constrained image splicing detection and localization based on atrous convolution," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2551–2566, Oct. 2019.

[3] K. Bahrami, A. C. Kot, L. Li, and H. Li, "Blurred image splicing localization by exposing blur type inconsistency," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 5, pp. 999–1009, May 2015.

[4] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," 2019, *arXiv:1910.08854*. [Online]. Available: http://arxiv.org/abs/1910.08854

[5] Z. Fan and R. L. D. Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 230–235, Feb. 2003.

[6] L. Dou, Z. Qian, C. Qin, G. Feng, and X. Zhang, "Anti-forensics of diffusion-based image inpainting," *J. Electron. Imag.*, vol. 29, no. 4, Aug. 2020, Art. no. 043026.

[7] J. Wu and W. Sun, "Towards multi-operation image anti-forensics with generative adversarial networks," *Comput. Secur.*, vol. 100, Jan. 2021, Art. no. 102083.

[8] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-forensics of JPEG compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 1694–1697.

[9] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "The cost of JPEG compression anti-forensics," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 1884–1887.

[10] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "JPEG anti-forensics with improved tradeoff between forensic undetectability and image quality," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1211–1226, Aug. 2014.

[11] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 576–584.

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

[13] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik, "Compression artifacts removal using convolutional neural networks," 2016, *arXiv:1605.00366*. [Online]. Available: http://arxiv.org/abs/1605.00366

[14] J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-46448-0_38.

[15] Y. Luo, H. Zi, Q. Zhang, and X. Kang, "Anti-forensics of JPEG compression using generative adversarial networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 952–956.

[16] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.

[17] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Information Hiding. IH* (Lecture Notes in Computer Science), vol. 6958, T. Filler, T. Pevný, S. Craver, and A. Ker, Eds. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-642-24178-9_5.

[18] P. Bas and T. Furon. (2007). *Bows-2 (2007)*. [Online]. Available: http://bows2.gipsa-lab.inpg.fr

[19] W. Luo, J. Huang, and G. Qiu, "JPEG error analysis and its applications to digital image forensics," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 480–491, Sep. 2010.

[20] S. Lai and R. Böhme, "Countering counter-forensics: The case of JPEG compression," in *Information Hiding. IH* (Lecture Notes in Computer Science), vol. 6958, T. Filler, T. Pevný, S. Craver, and A. Ker, Eds. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-642-24178-9_20.

[21] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of JPEG images: Breaking the F5 algorithm," in *Information Hiding. IH* (Lecture Notes in Computer Science), vol. 2578, F. A. P. Petitcolas, Ed. Berlin, Germany: Springer, 2003, doi: 10.1007/3-540-36415-3_20.

[22] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Revealing the traces of JPEG compression anti-forensics," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 2, pp. 335–349, Feb. 2013.

[23] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "A variational approach to JPEG anti-forensics," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3058–3062.

[24] C. Chen and Y. Q. Shi, "JPEG image steganalysis utilizing both intrablock and interblock correlations," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 3029–3032.

[25] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.

[26] H. Li, W. Luo, and J. Huang, "Countering anti-JPEG compression forensics," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 241–244.

[27] E. Y. Lam and J. W. Goodman, "A mathematical analysis of the DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[28] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Undetectable image tampering through JPEG compression anti-forensics," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2109–2112.

[29] G. Singh and K. Singh, "Improved JPEG anti-forensics with better image visual quality and forensic undetectability," *Forensic Sci. Int.*, vol. 277, pp. 133–147, Aug. 2017.

[30] B. Li, Y. Q. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *Proc. IEEE 10th Workshop Multimedia Signal Process.*, Oct. 2008, pp. 730–735.

[31] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognit.*, vol. 42, no. 11, pp. 2492–2501, Nov. 2009.

[32] Q. Wang and R. Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP J. Inf. Secur.*, vol. 2016, no. 1, p. 23, Dec. 2016.

[33] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro, "Aligned and non-aligned double JPEG detection using convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 49, pp. 153–163, Nov. 2017.

[34] J. Park, D. Cho, W. Ahn, and H.-K. Lee, "Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 636–652.

[35] W. Ahn, S. H. Nam, M. Son, H. K. Lee, and S. Choi, "End-to-end double JPEG detection with a 3D convolutional network in the DCT domain," *Electron. Lett.*, vol. 56, no. 2, pp. 82–85, Jan. 2020.

[36] P. Sutthiwan and Y. Q. Shi, "Anti-forensics of double JPEG compression detection," in *Digital Forensics and Watermarking. IWDW* (Lecture Notes in Computer Science), vol. 7128, Y. Q. Shi, H. J. Kim, and F. Perez-Gonzalez, Eds. Berlin, Germany: Springer, 2012, doi: 10.1007/978-3-642-32205-1_33.

[37] H. Li, W. Luo, and J. Huang, "Anti-forensics of double JPEG compression with the same quantization matrix," *Multimedia Tools Appl.*, vol. 74, no. 17, pp. 6729–6744, Sep. 2015.

[38] X. Zhang, W. Yang, Y. Hu, and J. Liu, "DmCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 390–394.

[39] N. Kwak, J. Yoo, and S.-H. Lee, "Image restoration by estimating frequency distribution of local patches," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6684–6692.

[40] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep generative adversarial compression artifact removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4826–4835.

[41] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 2131–2145, Aug. 2019.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[43] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9906, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-46475-6_43.

[44] F. Alter, S. Durand, and J. Froment, "Adapted total variation for artifact free decompression of JPEG images," *J. Math. Imag. Vis.*, vol. 23, no. 2, pp. 199–211, Sep. 2005.

[45] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.

DOHYUN KIM received the B.S. degree from the Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology (KAIST), South Korea, the double major degree from the School of Computing Department, KAIST, in 2019, where he is currently pursuing the M.S. degree. His research interests include multimedia forensics, computer vision, image reconstruction, and machine learning.

WONHYUK AHN received the B.S. degree in software and computer engineering from Ajou University, South Korea, in 2016. He is currently pursuing the Ph.D. degree with the School of Computing, Korea Advanced Institute of Science and Technology (KAIST). His research interests include multimedia forensics, computer vision, and machine learning.

HEUNG-KYU LEE received the B.S. degree in electronics engineering from Seoul National University, Seoul, South Korea, in 1978, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 1981 and 1984, respectively. Since 1986, he has been a Professor with the School of Computing, KAIST, as well as a Technical Director with Digital Innotech Company. He has authored/coauthored over 200 international journal and conference papers. His major research interests include digital watermarking, digital fingerprinting, digital rights management, information hiding, and multimedia forensics. He has been a Reviewer of many international journals, including the *Journal of Electronic Imaging*, *Real-Time Imaging*, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.