

ROBUST IMAGE IDENTIFICATION FOR DOUBLE-COMPRESSED AND RESIZED JPEG IMAGES

Kenta Iida* and Hitoshi Kiya*

* Tokyo Metropolitan University

E-mail: iken729@gmail.com, kiya@tmu.ac.jp

Abstract—In the case that images are shared via social networking services (SNS) and cloud photo sharing services (CPSS), it is known that the JPEG images uploaded to the services are often re-compressed and resized by the providers. Because of such a situation, a new image identification scheme for double-compressed JPEG images having different sizes from that of a single-compressed one is proposed in this paper. The aim is to detect a single-compressed image that has the same original image as the double-compressed ones, even when the sizes of those compressed images are different. In the proposed scheme, a feature extracted from only DC coefficients in DCT coefficients is used for the identification. The use of the feature allows us not only to robustly avoid errors caused by double-compression but also to perform the identification for different size images. The simulation results demonstrate the effectiveness of the proposed one in terms of the querying performance.

I. INTRODUCTION

The growing popularity of photo sharing applications on the Internet has opened new perspectives in many research fields, including the emerging area of multimedia forensics. Those applications include social network services (SNS) like Facebook and cloud photo sharing services (CPSS) like Google photos. The huge amount of images uploaded to SNS and CPSS are generally stored in a compressed format as JPEG images, after being resized and re-compressed using different compression parameters from those used for the uploaded images [1]–[3]. Due to a such situation, identifying JPEG images which have the same original image has been required.

Several identification schemes and robust image hashing ones have been proposed to consider the relationship between images [4]–[17]. They have been developed for the various purposes: producing evidence regarding image integrity, robust image retrieval, finding illegally distributed images and so on. The conventional schemes for identifying images can be broadly classified into two types: compression-method-dependent and compression-method-independent. Compression-method-independent schemes include image retrieval and image hashing-based ones [14]–[17]. These schemes generally extract features from resized or divided images after decoding images, and then the features are converted to other representations. For instance, ITQ-based scheme [17] converts Gist descriptors [18] generated from divided images. The compression-method-independent schemes have tried not

only to identify images having the different sizes but also to consider several noises including errors caused by lossy compression. However, they sometimes miss detecting slight differences because they mainly aim to retrieve similar images.

On the other hand, due to the use of robust features against JPEG errors, compression-method-dependent schemes [7]–[13] have the stronger robustness than the first type ones. The schemes [7]–[10] use positive and negative signs of discrete cosine transform (DCT) coefficients, and the schemes [11]–[13] focus on the positions in which DCT coefficients have zero values. However, the identification for images having different sizes can not be performed. In addition, most of them do not consider the identification between single-compressed images and double-compressed ones.

Due to such situations, our proposed scheme can robustly identify JPEG images double-compressed under various compression conditions, even if the sizes of the images are different. The identification is carried out with a feature extracted from DC coefficients. The use of the DC coefficients-based feature allows us not only to avoid errors caused by double-compression but also to achieved that images having different sizes can be identified. The simulation results demonstrate that the proposed scheme enables to detect slight differences, even if images are very similar.

II. PRELIMINARIES

A. JPEG Encoding

The JPEG standard is the most widely used image compression standard. The JPEG encoding procedure can be summarized as follows.

- 1) Perform color transformation from RGB space to $YCbCr$ space and sub-sample C_b and C_r .
- 2) Divide an image into non-overlapping consecutive 8×8 -blocks.
- 3) Apply DCT to each block to obtain 8×8 DCT coefficients S , after mapping all pixel values in each block from $[0,255]$ to $[-128,127]$ by subtracting 128 in general.
- 4) Quantize S using a quantization matrix Q .
- 5) Entropy code it using Huffman coding.

A DC coefficient $S(0,0)$ in each block is obtained by the following equation, where $I(b_x, b_y)$ represents a level-shifted

pixel value at the position (b_x, b_y) in a block.

$$S(0, 0) = \frac{1}{8} \sum_{b_x=0}^7 \sum_{b_y=0}^7 I(b_x, b_y) \quad (1)$$

The range of the DC coefficient is $[-1024, 1016]$.

In step 4), a quantization matrix \mathbf{Q} with 8×8 components is used to obtain a matrix \mathbf{S}_q from \mathbf{S} . For example,

$$S_q(u, v) = \text{round} \left(\frac{S(u, v)}{Q(u, v)} \right), \quad 0 \leq u \leq 7, \quad 0 \leq v \leq 7, \quad (2)$$

where $S(u, v)$, $Q(u, v)$ and $S_q(u, v)$ represent the (u, v) element of \mathbf{S} , \mathbf{Q} and \mathbf{S}_q respectively. The $\text{round}(x)$ function is used to round a value x to the nearest integer value and $\lfloor x \rfloor$ denotes the integer part of x .

The quality factor QF ($1 \leq QF \leq 100$) parameter is used to control a matrix \mathbf{Q} . The large QF results in a high quality image.

B. Image Manipulation by SNS/CPSS Provider

Let us consider that JPEG images are uploaded to a SNS/CPSS provider. It is known that JPEG images uploaded to SNS providers are often manipulated as below [1]–[3].

- Editing metadata and filenames
Most of metadata in the header are deleted for privacy-concerns and the filenames of uploaded images are changed.
- Re-compressing uploaded images
Before stored in a cloud storage, uploaded images are decoded once and then the images are compressed again under the different coding condition.
- Resizing uploaded images
If uploaded images satisfy certain conditions, those images are resized. For instance, in Twitter, when the filesize of images is larger than 3MB or the size of images is larger than 4096×4096 , the images will be resized.

As well as SNS providers, CPSS providers also manipulate uploaded images. For instance, images uploaded to “Google photos” are often re-compressed and resized.

In order to identify images uploaded to SNS/CPSS, it is required that the re-compression and resizing are considered in identification schemes, although conventional compression-dependent-schemes consider only the re-compression.

C. Scenario

Let us consider a situation in which there are two or more compressed images generated under different or the same coding conditions. They originated from the same image and were compressed under the various coding conditions. We refer to the identification of those images as “image identification”. Note that the aim of the image identification is not to retrieve visually similar images.

The scenario of this paper is illustrated in Fig. 1. In this scenario, a client/user identifies images by using an identification tool. When the client/user uploads JPEG images to SNS/CPSS, the features of these images are enrolled (extracted

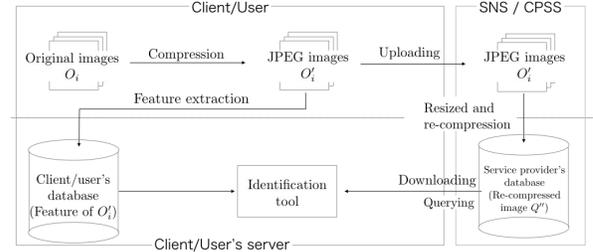


Fig. 1. Scenario

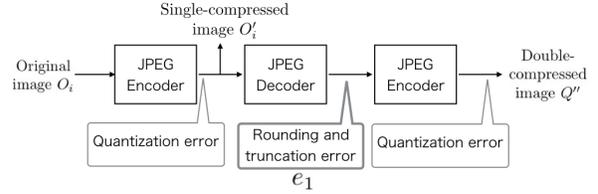


Fig. 2. JPEG errors in single-/double-compression

and then stored) in a client/user’s database. The uploaded images are resized to smaller sizes and re-compressed under different coding parameters, and then are stored in the cloud storage. Finally, the client/user carries out the identification after extracting the feature from a query image i.e. a downloaded image.

The JPEG standard is generally used as a lossy compression method, so several errors are caused in the generation process of double-compressed images [13], [19], as shown in Fig.2. In addition to “quantization error” in the encoding process, “rounding and truncation error” i.e. e_1 is caused in the decoding process. In the proposed scheme, the errors in both processes are considered to identify double-compressed images.

D. Notations and Terminologies

The notations and terminologies used in the following sections are listed here.

- O_i' represents a single-compressed image of an original image O_i .
- Q'' represents a double-compressed query image.
- M represents the number of 8×8 -blocks in an image.
- $O_i'(m)$ and $q''(m)$ indicate quantized DC coefficients in m th block in images O_i' and Q'' respectively ($0 \leq m < M$).
- $X_{O_i'}$ and $Y_{O_i'}$ represent the width and the height of O_i' respectively. As well, $X_{Q''}$ and $Y_{Q''}$ represent the width and the height of Q'' respectively.
- $Q_{O_i',L}$ and $Q_{Q'',L}$ indicate the DC components in the luminance quantization matrices, which are used to generate images O_i' and Q'' respectively.
- $QF_{O_i'}$ and $QF_{Q''}$ indicate quality factors used to generate O_i' and Q'' respectively.

- $\text{sgn}(a)$ represents the sign of a real value a as

$$\text{sgn}(a) = \begin{cases} 1, & a > 0, \\ 0, & a = 0, \\ -1, & a < 0. \end{cases} \quad (3)$$

III. PROPOSED IDENTIFICATION SCHEME

The proposed identification scheme aims to identify double-compressed images. In the proposed scheme, a feature of a JPEG image is extracted from only DC coefficients of Y component. Although the identification scheme explained in this section assumes the identification for the same size images, this scheme is easily extended for the identification of the different size images. The enrollment and identification processes are performed as below.

1) Enrollment Process

In order to enroll image O'_i as the feature vector $\mathbf{v}_{O'_i} \in \mathbb{R}^{\lceil \frac{X_{O'_i}}{8} \rceil * \lceil \frac{Y_{O'_i}}{8} \rceil * 1}$, a client/user carries out the following steps.

- Set values M , th and Δ , where th and Δ represent a threshold value and a parameter used for the feature extraction.
- Set $m := 0$.
- Extract a component of the feature vector $v_{O'_i}(m)$ from a DC coefficient $O'_i(m)$ as

$$v_{O'_i}(m) = \begin{cases} 0, & -th \leq O'_i(m) \leq th, \\ \text{round}\left(\frac{Q_{O'_i,L} * O'_i(m)}{\Delta}\right) \\ + \text{sgn}(O'_i(m)), & \text{otherwise,} \end{cases} \quad (4)$$

where $v_{O'_i}(m)$ represents the m th component of the feature $\mathbf{v}_{O'_i}$.

- Set $m := m + 1$. If $m < M$, return to step (c). Otherwise, store $\mathbf{v}_{O'_i}$ as the feature in the client/user's database.

For the feature extraction, a threshold value th and a parameter Δ are used. The aim of using th is to avoid the effect of double-compression i.e. e_1 , and Δ determines the amount of feature data stored in the database. Figure 3 shows the relation between a DC coefficient $O'_i(m)$ and a component of the feature $v_{O'_i}(m)$. As shown in Fig.3(a), when $\Delta > 2048$, each component of the feature has one of three values, i.e., -1, 0, 1. On the other hand, the component of the feature generated with $\Delta \leq 2048$ has more various values (see in Fig.3(b)). As shown in Fig.3, $\tau = th * Q_{O'_i,L}$ is a parameter to control robustness against errors caused by double-compression.

2) Identification Process

In order to compare image Q'' with image O'_i , the client/user carries out the following steps.

- Set values M , th , Δ , $d_{O'_i}$ and $d_{Q''}$, where $d_{O'_i}$ and $d_{Q''}$ are parameters for the identification. It is required that the parameters M , th and Δ are the same as those selected in step (a) of the enrollment process.
- Set $m := 0$.

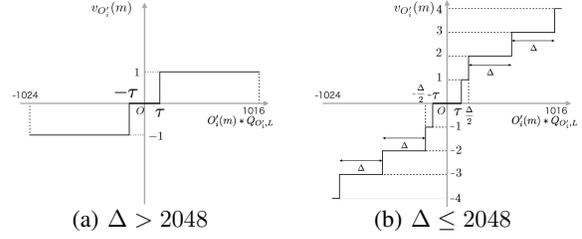


Fig. 3. Examples of the relation between $O'_i(m)$ and $v_{O'_i}(m)$, where $\tau = th * Q_{O'_i,L}$

- Extract a component of the feature $v_{Q''}(m)$ from a DC coefficient $q''(m)$ as

$$v_{Q''}(m) = \begin{cases} 0, & -th \leq q''(m) \leq th, \\ \text{round}\left(\frac{Q_{Q'',L} * q''(m)}{\Delta}\right) \\ + \text{sgn}(q''(m)), & \text{otherwise.} \end{cases} \quad (5)$$

- If $|v_{O'_i}(m)| > d_{O'_i}$ or $|v_{Q''}(m)| > d_{Q''}$, proceed to step (f).
- If $\text{sgn}(v_{O'_i}(m)) \neq \text{sgn}(v_{Q''}(m))$, the client/user judges that O'_i and Q'' are generated from different original images and the process for image O'_i is halted.
- Set $m := m + 1$. If $m < M$, return to step (c). Otherwise, the client/user judges that O'_i and Q'' are generated from the same original image.

As shown above, by using the feature extracted from DC coefficients, the identification is carried out in the proposed scheme. The following are the reasons why this feature is used in this paper.

- To identify different size images
In the case of using only DC coefficients mapped with Δ , as shown in Sec. IV, DC coefficients in the resized image can be calculated from ones in the image before resizing.
- To determine parameters independently of the size of images
The conventional scheme for double-compressed images [13], which uses not only DC but also AC coefficients, requires the setting of the parameter related to the size of identified images. On the other hand, the parameters used in the proposed scheme, i.e. th and Δ are independent of the size of images.

In addition to these advantages, the use of th allows us to reduce the influence of errors caused by the double-compression.

$d_{O'_i}$ and $d_{Q''}$ are required for the identification for different size images.

IV. FEATURE MODIFICATION FOR IDENTIFICATION OF DIFFERENT SIZE IMAGES

In the processes mentioned above, the identification for the same size images is assumed. However, images uploaded to

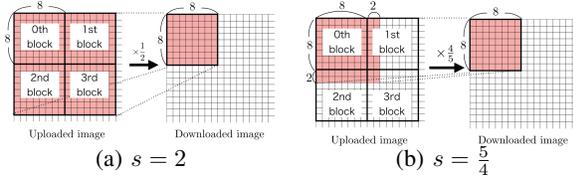


Fig. 4. Examples of the relationship between uploaded images and downloaded images with resizing

SNS/CPSS providers are sometimes resized as smaller images. Therefore, in order to identify different size images in the process mentioned in Sec. III, a modification method for the feature stored in the database is proposed in this section.

A. Strategy for Resized Images

Let us consider that an uploaded image with the size of $Y \times X$ is resized to $\frac{1}{s}$ times size, i.e. $\lceil \frac{Y}{s} \rceil \times \lceil \frac{X}{s} \rceil$, where s is a positive value and $\lceil a \rceil$ represents the ceiling of a real value a . As shown in Fig.4 (a), when the size of an uploaded image is changed to the half, i.e. $s = 2$, 0th block in the downloaded image is computed by using four blocks from 0th block to 3rd block in the uploaded image. The DC coefficient of every block $S(0,0)$ is defined by Eq.(1), so DC coefficients in the downloaded image is estimated by calculating the average of the corresponding DC coefficients in the uploaded image as

$$q''(0) = \frac{1}{4}O'_i(0) + \frac{1}{4}O'_i(1) + \frac{1}{4}O'_i(2) + \frac{1}{4}O'_i(3). \quad (6)$$

When s is not an integer value as shown in Fig.4 (b), the weighted average values of DC coefficients should be calculated, based on the number of corresponding pixels of each block in the uploaded image. For instance, as shown in Fig.4 (b), i.e. for $s = \frac{5}{4}$, the weights of four blocks are $\frac{64}{100}, \frac{16}{100}, \frac{16}{100}, \frac{4}{100}$ respectively. Therefore, $q''(0)$ is estimated by

$$q''(0) = \frac{64}{100}O'_i(0) + \frac{16}{100}O'_i(1) + \frac{16}{100}O'_i(2) + \frac{4}{100}O'_i(3). \quad (7)$$

Note that $10 \times 10 = 100$ pixels in an uploaded image are reduced to $8 \times 8 = 64$ pixels in the downloaded image for $s = \frac{5}{4}$.

In the practical, the estimated feature matrix \mathbf{D} is computed by using the feature matrix \mathbf{U} reproduced from the feature vector of an uploaded image $\mathbf{v}_{O'_i}$, where $\mathbf{U} \in \mathbb{R}^{\lceil \frac{X_{O'_i}}{s} \rceil \times \lceil \frac{Y_{O'_i}}{s} \rceil}$ and $\mathbf{D} \in \mathbb{R}^{\lceil \frac{X_{Q''}}{s} \rceil \times \lceil \frac{Y_{Q''}}{s} \rceil}$. In the examples in Fig.4, the values in Eq.(7) can be expressed as

$$D(0,0) = \frac{64}{100}U(0,0) + \frac{16}{100}U(1,0) + \frac{16}{100}U(0,1) + \frac{4}{100}U(1,1), \quad (8)$$

where $U(x_{O'_i}, y_{O'_i})$ is the $(x_{O'_i}, y_{O'_i})$ component of \mathbf{U} ($0 \leq x_{O'_i} < \lceil \frac{X_{O'_i}}{s} \rceil, 0 \leq y_{O'_i} < \lceil \frac{Y_{O'_i}}{s} \rceil$), and \mathbf{U} is mapped from $\mathbf{v}_{O'_i}$.

B. Modification of Enrolled Feature

According to the strategy mentioned above, when the size of query images is not the same as that of the uploaded image, the enrolled features are modified before the identification process. The modification process is shown as below.

- Set values $X_{O'_i}, Y_{O'_i}, X_{Q''}$ and $Y_{Q''}$.
- Map $\mathbf{v}_{O'_i}$ into a matrix \mathbf{U} .
- Set an estimated feature matrix \mathbf{D} as a zero matrix.
- Calculate dx and dy as below.

$$dx = \frac{8X_{O'_i}}{X_{Q''}}, \quad dy = \frac{8Y_{O'_i}}{Y_{Q''}}. \quad (9)$$

- Set $x := 0, y := 0, x_{Q''} := 0$ and $y_{Q''} := 0$.
- Set $x_{O'_i} := 0$ and $y_{O'_i} := 0$.
- Calculate a component of the weight matrix $\mathbf{W} \in \mathbb{R}^{\lceil \frac{X_{O'_i}}{s} \rceil \times \lceil \frac{Y_{O'_i}}{s} \rceil}$ by

$$W(x_{O'_i}, y_{O'_i}) = \sum_{x_I=x_{O'_i}*8}^{x_{O'_i}*8+7} \sum_{y_I=y_{O'_i}*8}^{y_{O'_i}*8+7} \frac{Z(x_I, y_I)}{dx * dy}, \quad (10)$$

where x_I and y_I are integer values and

$$Z(x_I, y_I) = \begin{cases} 1, & x \leq x_I < x + dx \\ & \text{and } y \leq y_I < y + dy, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

- Update $D(x_{Q''}, y_{Q''})$ by

$$D(x_{Q''}, y_{Q''}) = D(x_{Q''}, y_{Q''}) + W(x_{O'_i}, y_{O'_i}) * U(x_{O'_i}, y_{O'_i}). \quad (12)$$

- Set $x_{O'_i} := x_{O'_i} + 1$. If $x_{O'_i} < \lceil \frac{X_{O'_i}}{s} \rceil$, return to step (g).
- Set $x_{O'_i} := 0$ and $y_{O'_i} := y_{O'_i} + 1$. If $y_{O'_i} < \lceil \frac{Y_{O'_i}}{s} \rceil$, return to step (g).
- Set $x_{O'_i} := 0, y_{O'_i} := 0, x_{Q''} := x_{Q''} + 1$ and $x := x + dx$. If $x + dx - 1 < X_{Q''}$, return to step (f).
- Set $x_{Q''} := 0, x := 0, y_{Q''} := y_{Q''} + 1$ and $y := y + dy$. If $y + dy - 1 < Y_{Q''}$, return to step (f).
- Map the estimated feature matrix \mathbf{D} into a feature vector of the download image $\mathbf{v}_D \in \mathbb{R}^{\lceil \frac{X_{Q''}}{s} \rceil * \lceil \frac{Y_{Q''}}{s} \rceil * 1}$.

After this modification, the identification process in Sec.III 2) can be carried out by replacing $\mathbf{v}_{O'_i}$ with \mathbf{v}_D . Actually, there is the estimation error, so that $d_{O'_i}$ and $d_{Q''}$ are used for avoiding this error.

The use of the feature extracted from DC coefficients allows us not only to avoid the errors in double-compression but also to identify the different size images. The effectiveness of the proposed scheme will be shown in Sec.V.

V. SIMULATION

A number of simulations were conducted to evaluate the performance of the proposed scheme. We used the encoder and the decoder from the IJG (Independent JPEG Group) in the simulations [20].

A. Selection of Threshold Value and Parameters

In order to select the values of th , Δ , $d_{O'_i}$ and $d_{Q''}$, we conducted preliminary experiments as shown below.

1). Determination of th

- (a) Data set
885×6 single-compressed images were generated from 885 original images in Uncompressed Color Image Database (UCID) [21] with six quality factors ($QF = 70, 75, 80, 85, 90, 95$), and then every single-compressed one was re-compressed with six quality factors ($QF = 70, 75, 80, 85, 90, 95$) to obtain 885×6×6 double-compressed images.
- (b) Selection of single-compressed image
One single-compressed image was selected from 885×6 single-compressed ones.
- (c) Comparison of DC coefficients
At first, a double-compressed image was selected from six double-compressed images generated from the selected single-compressed one. Next, for all DC coefficients of two the selected JPEG images, the relation at the same block position between the two images was investigated. When both DC coefficients in a block position have no zero value and the signs of the DC values are different, a larger DC absolute value in the block was stored. Accordingly, all larger DC ones at the blocks at which the above condition was satisfied were stored. This process was conducted for six corresponding double-compressed images.

Step (b) and step (c) were carried out until all single-compressed images were selected in step (b).

- 4) Selection of th
The largest absolute value in the stored ones was chosen as th .

According to the above procedure, th was experimentally determined as 14. The parameter th is used to skip small DC coefficients at step (d) of the identification process, because the signs of such coefficients are easily inverted by the effect of double-compression.

2). Determination of Δ

After step (a) and step (b) as mentioned above, the following steps were conducted.

- (c) Calculate the differences between DC coefficients
At first, a double-compressed image was selected from six double-compressed images generated from the selected single-compressed one. Next, for all DC coefficients of two the selected images, the relation of DC coefficients at each position was investigated. When both DC coefficients at each position had the same sign and larger absolute values than th , the absolute value of the difference between two the DC values was saved for all positions at which the condition was satisfied, respectively. This process was conducted for six corresponding double-compressed images.



Fig. 5. Examples of test images in HPID (288×384)

Step (b) and step (c) were carried out until the all single-compressed images were selected in step (b).

- (d) Selection of Δ
The largest absolute value in the stored ones was selected as Δ .

By conducting the above steps, Δ was determined as 50.

3). Determination of $d_{O'_i}$ and $d_{Q''}$

Using th and Δ selected by the above procedures, the identification experiments were performed by using various JPEG images while changing the values of $d_{O'_i}$ and $d_{Q''}$. From the result, $d_{O'_i}$ and $d_{Q''}$ were determined as 4.

As shown in simulation results later, the use of the parameters $th = 14$, $\Delta = 50$, $d_{O'_i} = 4$ and $d_{Q''} = 4$ provided a high performance, so this selection was good one, although other selections provided almost the same results.

B. Querying Performance

Next, we used the images in Head Pose Image Database (HPID) [22] to evaluate querying performance. As shown in Fig. 5, HPID consists of very similar images, where the size of images is 288×384. The main reason of using HPID is to show that the proposed scheme can detect a slight differences between the images. Therefore, we used 186 images of "Person01" in HPID as original images.

The proposed scheme was compared with two compression-method-dependent-schemes (zero positions-based scheme [13] and DC signs-based scheme [8]) and three image hashing-based schemes (low-rank and sparse decomposition-based scheme [15], quaternion-based one [16] and iterative quantization (ITQ)-based one [17]), where ITQ-based hash values were generated from 512 dimensional GIST feature vectors and each hash value was represented by 512 bits. In the schemes [15]–[17], the hamming distances between the hash value of a query image and those of all images in each database were calculated, and then images that had the smallest distance were chosen as the images generated from the same original image as the query, after decompressing all images.

1) Querying Performance for Images without Resizing

At first, the querying performances for images without resizing were evaluated. Table I summarizes the quality factors used to generate JPEG images, where DB_1 , DB_2 and DB_3 indicate the databases of client/user in Fig.1. First of all, 186 single-compressed images were generated from 186 original images for each database, so 558 single-compressed images were generated from original ones for three databases. Next, those single-compressed images were re-compressed with four

TABLE I
QUALITY FACTORS USED TO GENERATE JPEG IMAGES. DB_1 , DB_2 AND DB_3 INDICATE DATABASES OF CLIENT/USER IN FIG.1

JPEG images		Quality factors
Uploaded images	DB_1	$QF_{O'_i} = 95$
	DB_2	$QF_{O'_i} = 85$
	DB_3	$QF_{O'_i} = 75$
Downloaded images (Query images Q'')		$QF_{Q''} = 71, 75, 80, 85$

TABLE II
QUERYING PERFORMANCES FOR IMAGES IN HPID

scheme	database	Precision[%]	Recall[%]
proposed ($\Delta = 50, d_{O'_i} = 4, d_{Q''} = 4$)	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
DC signs [8]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
zero value positions [13]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
low-rank and sparse decomposition [15]	DB_1	97.21	98.39
	DB_2	98.41	99.73
	DB_3	96.35	99.33
quaternion [16]	DB_1	99.60	100
	DB_2	99.60	99.87
	DB_3	100	100
ITQ [17]	DB_1	67.24	99.33
	DB_2	67.67	99.87
	DB_3	62.98	98.79

quality factors i.e. $QF_{Q''} = 71, 75, 80, 85$, where 744 double-compressed images were generated for each database. It is known that the range of quality factors used for re-compression in SNS is [71,85] as in [3], so these quality factors were used. Thus, to confirm whether each query image has the same original image as one of 186 single-compressed images, 186×744 identification operations were carried out for each database. For instance, in order to perform the identification operations for DB_1 , features were extracted from 186 single-compressed images compressed with $QF_{O'_i} = 95$, and 186 images with $QF_{O'_i} = 95$ were re-compressed with $QF_{Q''} = 71, 75, 80, 85$ respectively to generate 744 query images for DB_1 .

Table II shows *Precision* and *Recall*, defined by

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TP}{TP + FN}, \quad (13)$$

where TP, FP and FN represent the number of true positive, false positive and false negative matches respectively. Note that *Recall* = 100[%] means that there were no false negative matches, and *Precision* = 100[%] means that there were no false positive matches.

It is confirmed that the proposed scheme and two-compression-method-dependent ones achieved both *Recall* = 100[%] and *Precision* = 100[%], although the image hashing-based ones did not.

2) Querying Performances for Images with Resizing

Next, the querying performances for images with resizing were evaluated. The images stored as the feature and query images were generated by following the conditions shown in Tab. III. For instance, after 186 original images were resized

TABLE III
SIZES OF RESIZED IMAGES AND QUALITY FACTORS USED TO GENERATE JPEG IMAGES. DB_4 , DB_5 , DB_6 , DB_7 , DB_8 AND DB_9 INDICATE DATABASES OF CLIENT/USER IN FIG.1

JPEG images		Size	Quality factors
Uploaded images	DB_4	960×1280	$QF_{O'_i} = 95$
	DB_5	960×1280	$QF_{O'_i} = 85$
	DB_6	960×1280	$QF_{O'_i} = 75$
	DB_7	1440×1920	$QF_{O'_i} = 95$
	DB_8	1440×1920	$QF_{O'_i} = 85$
	DB_9	1440×1920	$QF_{O'_i} = 75$
Downloaded images (Query images Q'')		720×960	$QF_{Q''} = 71, 75, 80, 85$

to the size 960×1280 and compressed with $QF_{O'_i} = 95$, features stored in the database DB_4 were extracted from the generated images. In order to generate query images for DB_4 , the images enrolled as features were resized to the size 720×960, and then the resized images were compressed with $QF_{Q''} = 71, 75, 80, 85$.

Table IV shows the results, where “-” means that the identification can not be applied. The two compression-method-dependent schemes [8], [13] assume the identification for the same size images, so that they were not evaluated in this simulation. It is confirmed from Table IV that only the querying performances of the proposed scheme were perfect as well as the performances for the same size images. Therefore, the proposed scheme outperformed the conventional schemes as well as for images without resizing.

VI. CONCLUSION

In this paper, a new image identification scheme for double-compressed JPEG images was proposed to relate a query image with images uploaded to SNS/CPSS. The proposed scheme uses a feature extracted from DC coefficients in Y component. The use of the feature allows us to avoid the errors caused by double-compression. In addition, the identification for the different size images can be performed, although the conventional compression-method-dependent schemes can not. The simulation results showed that the proposed scheme detected slightly differences and outperformed other schemes including the state-of-art one, even if images were very similar. We plan to extend the proposed scheme as a tamper localization in our future work.

REFERENCES

- [1] R. Caldelli, R. Becarelli, and I. Amerini, “Image origin classification based on social network provenance,” *IEEE Trans. Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, 2017.
- [2] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato, “A classification engine for image ballistics of social data,” *Computing Research Repository*, vol. abs/1610.06347, 2016.
- [3] T. Chuman, K. Iida, and H. Kiya, “Image manipulation on social media for encryption-then-compression systems,” in *Proc. APSIPA Annual Summit and Conference*, 2017, pp. 858–863.
- [4] C.-Y. Lin and S.-F. Chang, “A robust image authentication method distinguishing jpeg compression from malicious manipulation,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 153–168, 2001.

TABLE IV
 QUERING PERFORMANCE FOR RESIZED IMAGES, WHERE “-” MEANS
 THAT THE IDENTIFICATION CAN NOT BE APPLIED.

scheme	database	Precision[%]	Recall[%]
proposed ($\Delta = 50, d_{Q'} = 4, d_{Q''} = 4$)	DB ₄	100	100
	DB ₅	100	100
	DB ₆	100	100
	DB ₇	100	100
	DB ₈	100	100
	DB ₉	100	100
DC Signs [8]	DB ₄	-	-
	DB ₅	-	-
	DB ₆	-	-
	DB ₇	-	-
	DB ₈	-	-
	DB ₉	-	-
zero value positions [13]	DB ₄	-	-
	DB ₅	-	-
	DB ₆	-	-
	DB ₇	-	-
	DB ₈	-	-
	DB ₉	-	-
low-rank and sparse decomposition [15]	DB ₄	97.37	99.60
	DB ₅	97.10	98.92
	DB ₆	95.50	99.73
	DB ₇	96.86	99.60
	DB ₈	97.49	99.19
	DB ₉	97.48	98.79
quaternion [16]	DB ₄	98.94	100
	DB ₅	99.73	99.73
	DB ₆	98.80	100
	DB ₇	99.73	99.73
	DB ₈	98.94	100
	DB ₉	98.94	100
ITQ [17]	DB ₄	72.65	94.62
	DB ₅	77.58	97.17
	DB ₆	75.53	96.64
	DB ₇	52.58	79.44
	DB ₈	54.05	78.09
	DB ₉	50.99	79.30

[5] Z. Fan and R. L. de Queiroz, “Identification of bitmap compression history: Jpeg detection and quantizer estimation,” *IEEE Trans. on Image Processing*, vol. 12, no. 2, pp. 230–235, 2003.

[6] K.O. Cheng, N.F. Law, and W.C. Siu, “A fast approach for identifying similar features in retrieval of jpeg and jpeg2000 images,” in *Proc. APSIPA Annual Summit and Conference*, 2009, pp. 258–261.

[7] F. Arnia, I. Iizuka, M. Fujiyoshi, and H. Kiya, “Fast and robust identification methods for jpeg images with various compression ratios,” in *Proc. IEEE Int’l Conf. on Acoustics Speech and Signal Processing Proceedings*, 2006, vol. 2, pp. II–II.

[8] K. Iida and H. Kiya, “Robust image identification for double-compressed jpeg images,” in *Proc. Int’l Conf. on Communications*, 2018, pp. 143–146.

[9] K. Iida and H. Kiya, “Secure and robust identification based on fuzzy commitment scheme for jpeg images,” in *Proc. Int’l Symposium on Broadband Multimedia Systems and Broadcasting*, 2016, pp. 1–5.

[10] K. Iida and H. Kiya, “Fuzzy commitment scheme-based secure identification for jpeg images with various compression ratios,” *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 99, no. 11, pp. 1962–1970, 2016.

[11] K. Iida and H. Kiya, “Robust image identification without visible information for jpeg images,” *IEICE Trans. on Inf. & Sys.*, vol. 101, no. 1, pp. 13–19, 2018.

[12] K. Iida and H. Kiya, “Robust image identification with secure features for jpeg images,” in *Proc. IEEE Int’l Conf. on Image Processing*, 2017, pp. 4342–4346.

[13] K. Iida and H. Kiya, “Robust image identification without any visible information for double-compressed jpeg images,” in *Proc. APSIPA Annual Summit and Conference*, 2017, vol. 2017, pp. 12–15.

[14] C. Lakovidou, N. Anagnostopoulos, A. Kapoutsis, Y. Boutalis, M. Lux, and S.A. Chatzichristofis, “Localizing global descriptors for content-based image retrieval,” *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 80, 2015.

[15] Y. Li and P. Wang, “Robust image hashing based on low-rank and

sparse decomposition,” in *Proc. IEEE Int’l Conf. on Acoustics, Speech and Signal Processing*, 2016, pp. 2154–2158.

[16] Y. N. Li, P. Wang, and Y. T. Su, “Robust image hashing based on selective quaternion invariance,” *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2396–2400, 2015.

[17] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[18] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[19] F. Huang, J. Huang, and Y.Q. Shi, “Detecting double jpeg compression with the same quantization matrix,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 848–856, 2010.

[20] “The independent jpeg group software jpeg codec,” <http://www.ijg.org/>.

[21] G. Schaefer and M. Stich, “Ucid: An uncompressed color image database,” in *Electronic Imaging 2004*, 2003, pp. 472–480.

[22] N. Gourier, D. Hall, and J. L. Crowley, “Estimating face orientation from robust detection of salient facial structures,” in *Proc. Int’l Workshop on Visual Observation of Deictic Gestures*, 2004, vol. 6.