PAPER *Special Section on Enriched Multimedia—Making Multimedia More Convenient and Safer—*

Robust Image Identification with DC Coefficients for Double-Compressed JPEG Images

Kenta IIDA[†a)], ***Member and*** **Hitoshi KIYA**[†b)], ***Fellow***

SUMMARY In the case that images are shared via social networking services (SNS) and cloud photo storage services (CPSS), it is known that the JPEG images uploaded to the services are mostly re-compressed by the providers. Because of such a situation, a new image identification scheme for double-compressed JPEG images 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. 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.

key words: *JPEG, image identification, social networks*

1. 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 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]–[15]. 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 [12]–[15]. 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 [15] converts Gist descriptors [16] 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]–[11] have the stronger robustness than the first type ones. The schemes [7], [8] use positive and negative signs of discrete cosine transform (DCT) coefficients, and the scheme [9]–[11] focuses on the positions in which DCT coefficients have zero values. They guarantee that there are no false negative matches for the identification between single-compressed images. However, the performances of the identification for double-compressed images degrade because the conventional schemes do not consider the errors caused in double-compression. Moreover, the identification for different size images has never been discussed in the compression-method-dependent schemes.

Due to such situations, our proposed scheme can robustly identify JPEG images double-compressed under various compression conditions. The identification is carried out with a feature extracted from DC coefficients. The use of the DC coefficients-based feature allows us to identify images with avoiding errors caused by double-compression. In addition, it is also 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.

2. Preliminaries

2.1 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 YC_bC_r 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 $\mathbf{S}$, after mapping all pixel values in each block from [0,255] to [-128,127] by subtracting 128 in general.

Manuscript received March 28, 2018.
Manuscript revised August 12, 2018.
Manuscript publicized October 19, 2018.

[†]The authors are with the Tokyo Metropolitan University, Hino-shi, 191–0065 Japan.
a) E-mail: iken729@gmail.com
b) E-mail: kiya@tmu.ac.jp

DOI: 10.1587/transinf.2018MUP0007

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.

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

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

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

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

where $S(u,v)$, $Q(u,v)$ and $S_q(u,v)$ represent the (u,v) element of **S**, **Q** and $\mathbf{S_q}$ respectively. The 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 $(1 \le QF \le 100)$ parameter is used to control a matrix **Q**. The large QF results in a high quality image.

2.2 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 file-size 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 these manipulations are considered in identification schemes.

2.3 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.

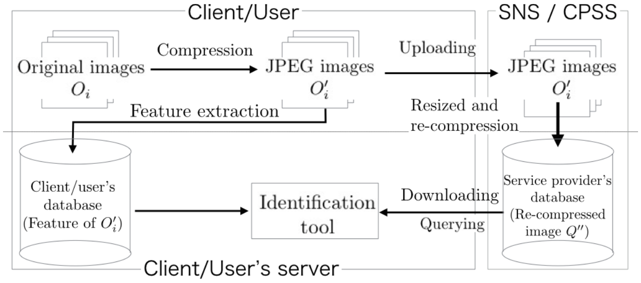


Fig. 1 Scenario

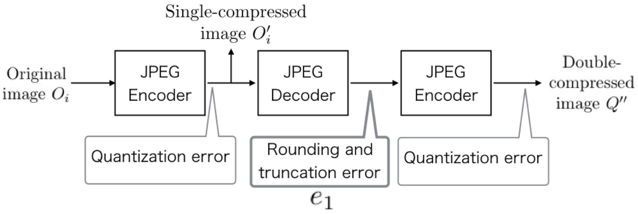


Fig. 2 JPEG errors in single-/double-compression

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 and then stored) in a client/user's database. The uploaded images are 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 [11], [17], 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.

2.4 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 mth block in images O'_i and Q'' respectively $(0 \le m < M)$.
- $X_{O'_i}$ 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.
- sgn(a) represents the sign of a real value a as

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

3. Proposed Identification Scheme

The proposed identification scheme aims to identify double-compressed images. The identification is performed with a feature extracted from DC coefficients. In this section, the relationship between the same size images is focused on.

3.1 Proposed Scheme

In the proposed scheme, a feature of a JPEG image is extracted from only DC coefficients of Y component. Enrollment and identification processes are explained, here.

1) Enrollment Process

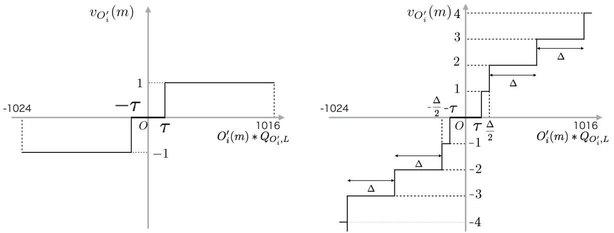
In order to enroll image O'_i as the feature vector $\mathbf{v}_{O'_i} \in \mathbb{R}^{\lceil \frac{X_{O'_i}}{8} \rceil * \lceil \frac{Y_{O'_i}}{8} \rceil \times 1}$, a client/user carries out the following steps.

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

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

where $v_{O'_i}(m)$ represents the mth component of the feature $\mathbf{v}_{O'_i}$.
(d) 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 robustness against quantization errors and 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, Δ and $\tau = th * Q_{O'_i,L}$ are parameters to control robustness against errors caused by double-compression.



(a) $\Delta > 2048$ (b) $\Delta \leq 2048$

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

2) Identification Process

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

(a) Set values M, th, Δ and d, where d is a parameter 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.
(b) Set $m := 0$.
(c) Extract a component of the feature $v_{Q''}(m)$ from a DC coefficient $q''(m)$ as

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

(d) If sgn($v_{O'_i}(m)$)=0 or sgn($v_{Q''}(m)$)=0, proceed to step (f).
(e) If sgn($v_{O'_i}(m)$) $\neq$ 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.
(f) If $|v_{O'_i}(m) - v_{Q''}(m)| > d$, the client/user judges that O'_i and Q'' are generated from different original images and the process for image O'_i is halted, where d is a parameter to determine an acceptance error value between features.
(g) 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.

The relation between $q''(m)$ and $v_{Q''}(m)$ in Eq. (5) is the same as the relation between $O'_i(m)$ and $v_{O'_i}(m)$ in Fig. 3. As shown above, by using the feature extracted from DC coefficients, the identification is carried out in the proposed scheme, although the conventional schemes use all DCT coefficients [9]–[11] or the signs of DCT coefficients [7], [8]. 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, as shown in

Sect. 4, 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 [11], 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* and Δ allows us to reduce the influence of errors caused by the double-compression.

4. Extension of Identification Scheme

Images uploaded to SNS/CPSS providers are sometimes resized as smaller images. Therefore, the proposed scheme is extended for the identification between different size images.

4.1 Strategy

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

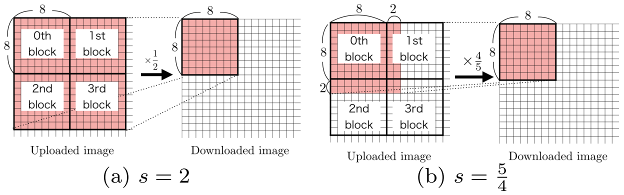
$$q''(0) = \frac{1}{4}O'_i(0) + \frac{1}{4}O'_i(1) + \frac{1}{4}O'_i(2) + \frac{1}{4}O'_i(3). \tag{6}$$

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

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

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 $\mathbf{U} \in \mathbb{R}^{\lceil \frac{X_{O'_i}}{8} \rceil \times \lceil \frac{Y_{O'_i}}{8} \rceil}$ and $\mathbf{D} \in \mathbb{R}^{\lceil \frac{X_{Q''}}{8} \rceil \times \lceil \frac{Y_{Q''}}{8} \rceil}$. In the examples in Fig. 4, the values in Eq. (7) can be expressed as



(a) $s = 2$ (b) $s = \frac{5}{4}$

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

$$\begin{aligned} D(0,0) =& \frac{64}{100}U(0,0) + \frac{16}{100}U(1,0) \\ &+ \frac{16}{100}U(0,1) + \frac{4}{100}U(1,1), \end{aligned} \tag{8}$$

where $U(x_{O'_i}, y_{O'_i})$ is the $(x_{O'_i}, y_{O'_i})$ component of $\mathbf{U}$ $(0 \leq x_{O'_i} < \lceil \frac{X_{O'_i}}{8} \rceil, 0 \leq y_{O'_i} < \lceil \frac{Y_{O'_i}}{8} \rceil)$, and $\mathbf{U}$ is mapped from $\mathbf{v}_{O'_i}$.

4.2 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.

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

$$dx = \frac{8X_{O'_i}}{X_{Q''}}, \; dy = \frac{8Y_{O'_i}}{Y_{Q''}}. \tag{9}$$

(e) Set $x := 0$, $y := 0$, $x_{Q''} := 0$ and $y_{Q''} := 0$.
(f) Set $x_{O'_i} := 0$ and $y_{O'_i} := 0$.
(g) Calculate a component of the weight matrix $\mathbf{W} \in \mathbb{R}^{\lceil \frac{X_{O'_i}}{8} \rceil \times \lceil \frac{Y_{O'_i}}{8} \rceil}$ by

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

where x_I and y_I are integer values and

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

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

$$\begin{aligned} D(x_{Q''}, y_{Q''}) =& D(x_{Q''}, y_{Q''}) \\ &+ W(x_{O'_i}, y_{O'_i}) * U(x_{O'_i}, y_{O'_i}). \end{aligned} \tag{12}$$

(i) Set $x_{O'_i} := x_{O'_i} + 1$. If $x_{O'_i} < \lceil \frac{X_{O'_i}}{8} \rceil$, return to step (g).
(j) Set $x_{O'_i} := 0$ and $y_{O'_i} := y_{O'_i} + 1$. If $y_{O'_i} < \lceil \frac{Y_{O'_i}}{8} \rceil$, return to step (g).
(k) Set $x_{O'_i} := 0$, $y_{O'_i} := 0$, $x_{Q''} := x_{Q''} + 1$ and $x := x + dx$.

If $x + dx - 1 < X_{O_i'}$, return to step (f).

(l) Set $x_{Q''} := 0$, $x := 0$, $y_{Q''} := y_{Q''} + 1$ and $y := y + dy$. If $y + dy - 1 < Y_{O_i'}$, return to step (f).

(m) Map the estimated feature matrix $\mathbf{D}$ into a feature vector of the download image $\mathbf{v}_D \in \mathbb{R}^{\lceil \frac{X_{Q''}}{8} \rceil * \lceil \frac{Y_{Q''}}{8} \rceil \times 1}$.

After this modification, the identification process in Sect. 3.1 2) can be carried out by replacing $\mathbf{v}_{O_i'}$ with $\mathbf{v}_D$.

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 Sect. 5.

5. 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 [18].

5.1 Selection of Parameters

In order to select the parameters *th*, Δ and *d*, we conducted preliminary experiments as shown below.

A. Determination of *th*

1) Data set
885×6 single-compressed images were generated from 885 original images in Uncompressed Color Image Database (UCID) [19] 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.
2) Selection of single-compressed image
One single-compressed image was selected from 885×6 single-compressed ones.
3) 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.

Steps 2) and 3) were carried out until all single-compressed images were selected in step 2).

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.

B. Determination of Δ

After step 1) and step 2) as mentioned above, the following steps were conducted.

3) 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.

Steps 2) and 3) were carried out until the all single-compressed images were selected in step 2).

4) 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.

C. Determination of *d*

Using *th* and Δ selected by the above procedures, the identification experiments were performed by using various JPEG images while changing the value of *d*. From the result, *d* was determined as 11.

As shown in simulation results later, the use of the parameters $th = 14$, $\Delta = 50$ and $d = 11$ provided a high performance, so this selection was good one, although other selections provided almost the same results. Normal users in SNS/CPSS can use these values for the identification as one of their choices.

5.2 Dataset

In order to evaluate querying performance between single-compressed images and double-compressed ones, some simulations were conducted in the case of using images in UKBench dataset [20] and Head Pose Image Database (HPID) [21] as original images respectively. UKBench dataset is usually used for evaluating the performance of image retrieval, and it consists of 10200 images (4 images per 2520 objects). In these images, we used 500 images, from No.00000 to No.00499. On the other hand, HPID contains face images of 15 persons and there are 186 images per person as shown in Fig. 6. The main reason of using HPID is

Fig. 5 Examples of test images in UKBench (480×640)

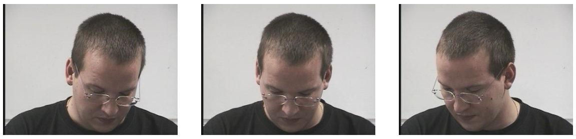


Fig. 6 Examples of test images in HPID (288×384)

to show that the proposed scheme can detect slight differences between images. Therefore, we used 186 images of "Person01" in HPID as original images.

5.3 Querying Performance for Images without Resizing

Table 1 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 and store the features of the single-compressed images. For instance, features stored in the database DB_1 were extracted from images compressed with $QF_{O'_i} = 95$. As query images for DB_1, images with $QF_{O'_i} = 95$ were re-compressed with $QF_{Q''} = 71, 75, 80, 85$ respectively. 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. Similarly, features in DB_2 and DB_3 were extracted from images with $QF_{O'_i} = 85$ and $QF_{O'_i} = 75$ respectively, and then images in each database were re-compressed with $QF_{Q''} = 71, 75, 80, 85$.

The proposed scheme was compared with two compression-method-dependent schemes (zero value positions-based scheme [9] and DCT signs-based one [7]) and three image hashing-based schemes (low-rank and sparse decomposition-based scheme [13], quaternion-based one [14] and iterative quantization (ITQ)-based one [15]), 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 [13]–[15], 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.

5.3.1 Performance for Images from UKBench

The querying performances were evaluated in the case of using 500 images in UKBench as original images. At first, 500 single-compressed images were generated from 500 original images for each database, so 1500 single-compressed images were generated from 500 original ones for three databases. Next, those single-compressed images were re-compressed with four quality factors i.e. $QF_{Q''} = 71, 75, 80, 85$, where 2000 double-compressed images were generated for each database, so there were totally 6000 double-compressed images as query images for three databases. Thus, to confirm whether each query image has the same original image as one of 500 single-compressed images, 500×2000 identification operations were carried out for each database.

Table 1 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 (single-compressed 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 2 Querying performance for images in UKBench

scheme	database	$Precision$[%]	$Recall$[%]
proposed ($\Delta = 50, d = 11$)	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
DCT signs [7]	DB_1	100	99.20
	DB_2	100	75.90
	DB_3	100	86.45
zero values positions [9]	DB_1	100	94.09
	DB_2	100	62.50
	DB_3	100	77.28
low-rank and sparse decomposition [13]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
quartenion [14]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
ITQ [15]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100

Table 2 shows $Precision$ and $Recall$, defined by

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

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.

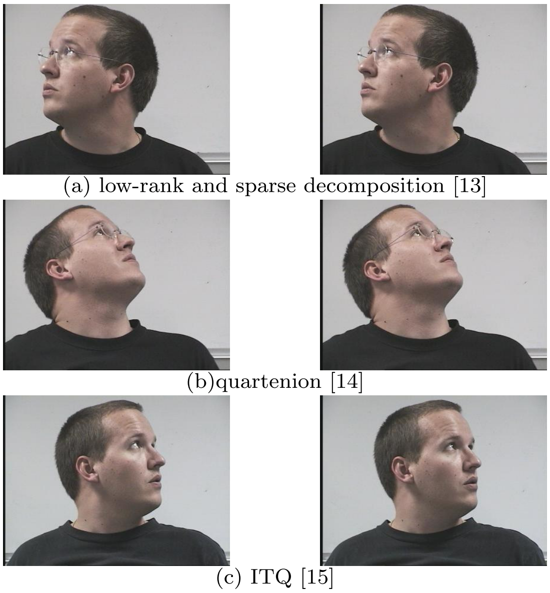
It is confirmed that proposed scheme achieved $Recall = 100\%$ for all databases, although the conventional compression-dependent schemes [7], [9] did not. The schemes [7], [9] guarantee $Recall = 100\%$ for single-compressed images in principle, however, it does not consider the errors caused by double-compression. As a result, the schemes degraded the querying performances for double-compressed images in terms of $Recall$.

5.3.2 Performance for Images from HPID

Next, we evaluated the querying performances for images from HPID. As well as for UKBench, JPEG images were generated from 186 original images under the conditions as shown in Table 1. At first, 186 single-compressed images

Table 3 Querying performances for images in HPID

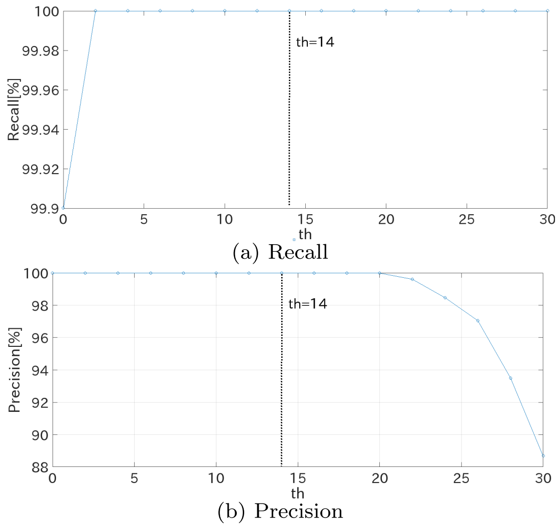
scheme	database	*Precision*[%]	*Recall*[%]
proposed ($\Delta = 50, d = 11$)	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
DCT signs [7]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
zero value positions [9]	DB_1	100	100
	DB_2	100	72.72
	DB_3	100	98.39
low-rank and sparse decomposition [13]	DB_1	97.21	98.39
	DB_2	98.41	99.73
	DB_3	96.35	99.33
quartenion [14]	DB_1	99.60	100
	DB_2	99.60	99.87
	DB_3	100	100
ITQ [15]	DB_1	67.24	99.33
	DB_2	67.67	99.87
	DB_3	62.98	98.79


(a) low-rank and sparse decomposition [13]

(b)quartenion [14]

(c) ITQ [15]

Fig. 7 Examples of couples of images incorrectly identified by the schemes [13]–[15]



(a) Recall

(b) Precision

Fig. 8 Recall and Precision values under $0 \leq th \leq 30$, $\Delta = 50$ and $d = 11$

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 quality factors i.e. $QF_{Q''} = 71, 75, 80, 85$, where 744 double-compressed images were generated for each database. 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.

It is confirmed from Table 3 that only the proposed and DCT signs-based schemes achieved both $Precision = 100\%$ and $Recall = 100\%$, i.e. perfect identification, although the other schemes did not. In the case of using the schemes [13]–[15], hash values of similar images are close, so that they provided some false negative and positive matches under the simulation conditions (Fig. 7 shows examples of incorrectly identified images). Therefore, the proposed scheme outperformed the conventional ones, even when two images were very similar.

5.3.3 Discussion

In the proposed scheme, three parameters th, Δ and d were used. Here, the effects of change of these parameters are discussed in the case of changing the values of these parameters. In order to evaluate the performances, the identification for UKBench images enrolled in DB_2 was performed under various values.

Figure 8 shows the identification results in the case that the value of th was changed, where $\Delta = 50$ and $d = 11$. When a small value was selected, i.e. $th < 2$, the value of *Recall* degraded because the signs of DC coefficients were inverted by the effect of double-compression. On the other hand, the use of a large value such as $th \geq 20$ decreased the value of *Precision* due to step (d) in the identification process. From Fig. 8, $th = 14$ was one of reasonable values for the identification.

Next, the effect of using different values as Δ and d under $th = 14$ was considered. It is confirmed from Fig. 9 that $d = 11$ and Δ=50 were one of the best selections under the experiment conditions.

We confirmed that the identification results had the same trends as those of the results in the case of the identification for HPID images and UKBench images enrolled in DB_1 and DB_3. The databases UKBench and HPID includes various images, so the proposed scheme would provide a high performance in many cases, although it is not guaranteed that these parameters were the best values for any images. Note that the users can select other values by following the way described in Sect. 5.1, if special images are used and users require a higher performance.

5.4 Querying Performance for Images with Resizing

Next, the performances for different size images were evaluated. The performances were compared with those of image hashing-schemes because the conventional compression-dependent schemes can not be applied for the identification of the different size images.

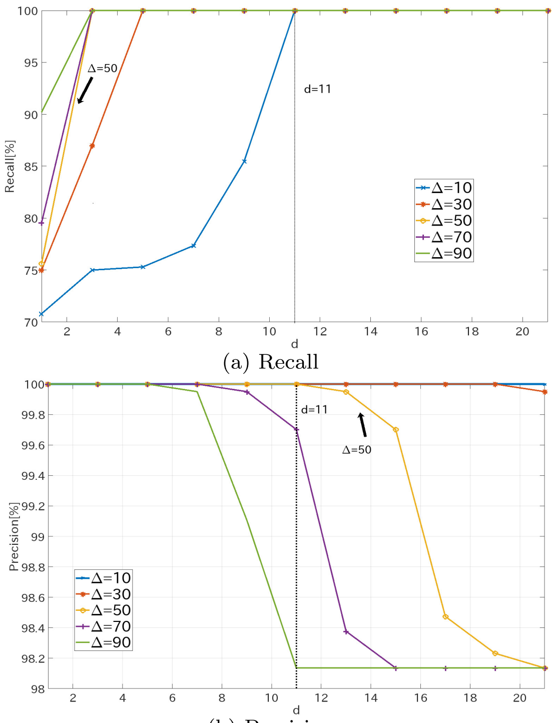
5.4.1 Performance for Different Size Images from UK-Bench

Identification performances for resized images in UKBench were evaluated here. As shown in Table 4, the images in UKBench were used after resizing. Note that uploaded images in DB_1, DB_2 and DB_3 were resized to $\frac{2}{3}$ times size, i.e. $s = \frac{3}{2}$ to generate query images. As well, ones in DB_4, DB_5 and DB_6 were resized to $\frac{1}{2}$ times size, i.e. $s = 2$. For instance, after 500 original images were resized to the size 960×1280 and compressed with $QF_{O'_i} = 95$, features stored in the database DB_1 were extracted from the generated images. To generate query images for DB_1, the images enrolled as features were resized to the size 720×960 and then the resized images were compressed with $QF_{Q''} = 71, 75, 80, 85$.

It is confirmed from Table 5 that the proposed scheme worked as well as the identification of the images having the same size.

5.4.2 Performance for Different Size Images from HPID

Under the coding conditions as well as in Sect. 5.4.1, the



(a) Recall

(b) Precision

Fig. 9 Recall and Presicion values under $th = 14$, $10 \leq \Delta \leq 90$ and $1 \leq d \leq 21$

Table 4 Sizes of images and quality factors used to generate JPEG images. Uploaded images in DB_1, DB_2 and DB_3 were resized to $\frac{2}{3}$ times size, i.e. $s = \frac{3}{2}$ to generate query images. As well, ones in DB_4, DB_5 and DB_6 were resized to $\frac{1}{2}$ times size, i.e. $s = 2$.

JPEG images		Size	Quality factors
Uploaded images (single-compressed images)	DB_1	960×1280	$QF_{O'_i} = 95$
	DB_2	960×1280	$QF_{O'_i} = 85$
	DB_3	960×1280	$QF_{O'_i} = 75$
	DB_4	1440×1920	$QF_{O'_i} = 95$
	DB_5	1440×1920	$QF_{O'_i} = 85$
	DB_6	1440×1920	$QF_{O'_i} = 75$
Downloaded images (Query images Q'')		720×960	$QF_{Q''} = 71, 75, 80, 85$

Table 5 Querying performance for resized images from UKBench

scheme	database	*Precision*[%]	*Recall*[%]
proposed ($\Delta = 50,\ d = 11$)	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
	DB_4	100	100
	DB_5	100	100
	DB_6	100	100
low-rank and sparse decomposition [13]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
	DB_4	100	100
	DB_5	100	100
	DB_6	100	100
quartenion [14]	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
	DB_4	100	100
	DB_5	100	100
	DB_6	100	100
ITQ [15]	DB_1	98.60	98.70
	DB_2	98.70	98.70
	DB_3	98.60	98.60
	DB_4	73.00	99.20
	DB_5	73.15	75.90
	DB_6	72.50	86.45

Table 6 Querying performance for resized images in HPID

scheme	database	*Precision*[%]	*Recall*[%]
proposed ($\Delta = 50,\ d = 11$)	DB_1	100	100
	DB_2	100	100
	DB_3	100	100
	DB_4	100	100
	DB_5	100	100
	DB_6	100	100
low-rank and sparse decomposition [13]	DB_1	97.37	99.60
	DB_2	97.10	98.92
	DB_3	95.50	99.73
	DB_4	96.86	99.60
	DB_5	97.49	99.19
	DB_6	97.48	98.79
quartenion [14]	DB_1	98.94	100
	DB_2	99.73	99.73
	DB_3	98.80	100
	DB_4	99.73	99.73
	DB_5	98.94	100
	DB_6	98.94	100
ITQ [15]	DB_1	72.65	94.62
	DB_2	77.58	97.17
	DB_3	75.53	96.64
	DB_4	52.58	79.44
	DB_5	54.05	78.09
	DB_6	50.99	79.30

performances for resized images of HPID were evaluated. The images stored as the feature and query images were generated following the conditions shown in Table 4. As shown in Table 6, the querying performances of the proposed scheme were perfect, while the other schemes degrade the querying performances in many cases. Therefore, the proposed scheme outperformed the conventional schemes as well as for images without resizing.

6. 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. Inf. Forensics Security, vol.12, no.6, pp.1299–1308, June 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,” Proc. APSIPA Annual Summit and Conference, pp.858–863, 2017.

[4] C.Y. Lin and S.F. Chang, “A robust image authentication method distinguishing jpeg compression from malicious manipulation,” IEEE Trans. Circuits Syst. Video Technol., vol.11, no.2, pp.153–168, Feb. 2001.

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

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

[8] K. Iida and H. Kiya, “Fuzzy commitment scheme-based secure identification for jpeg images with various compression ratios,” IEICE Trans. Fundamentals, vol.E99-A, no.11, pp.1962–1970, Nov. 2016.

[9] K. Iida and H. Kiya, “Robust image identification without visible information for jpeg images,” IEICE Trans. Inf. & Syst., vol.E101-D, no.1, pp.13–19, Jan. 2018.

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

[11] K. Iida and H. Kiya, “Robust image identification without any visible information for double-compressed jpeg images,” Proc. APSIPA Annual Summit and Conference, pp.852–857, 2017.

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

[13] Y. Li and P. Wang, “Robust image hashing based on low-rank and sparse decomposition,” Proc. IEEE Int’l Conf. on Acoustics, Speech Signal Process., pp.2154–2158, 2016.

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

[15] 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. Pattern Anal. Mach. Intell., vol.35, no.12, pp.2916–2929, Dec. 2013.

[16] 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, May 2001.

[17] F. Huang, J. Huang, and Y.Q. Shi, “Detecting double jpeg compression with the same quantization matrix,” IEEE Trans. Inf. Forensics Security, vol.5, no.4, pp.848–856, Dec. 2010.

[18] “The independent jpeg group software jpeg codec.” http://www.ijg.org/.

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

[20] “Ukbench dataset.” https://archive.org/details/ukbench.

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

Kenta Iida received his B.Eng. degree from Tokyo Metropolitan University, Japan in 2016. He is a Master course student at Tokyo Metropolitan University, Japan. His research interests include image processing, biometrics, and multimedia security. He is a student member of IEEE and IEICE.

Hitoshi Kiya received his B.Eng. and M.Eng. degrees from Nagaoka University of Technology, Japan, in 1980 and 1982, respectively, and his D.Eng. degree from Tokyo Metropolitan University in 1987. In 1982, he joined Tokyo Metropolitan University as an Assistant Professor, where he became a Full Professor in 2000. From 1995 to 1996, he attended the University of Sydney, Australia as a Visiting Fellow. He was/is the Chair of IEEE Signal Processing Society Japan Chapter, an Associate Editor for IEEE Trans. Image Processing, IEEE Trans. Signal Processing and IEEE Trans. Information Forensics and Security, respectively and Regional Director-at-Large for Region 10 of IEEE Signal processing Society. He also served as the President of IEICE Engineering Sciences Society (ESS), the Editor-in-Chief for IEICE ESS Publications, and a Vice President of APSIPA. He currently serves as the President-Elect of APSIPA. He is a Fellow of IEEE, IEICE and ITE.