Feature Extraction Suitable for Double JPEG Compression Analysis Based on Statistical Bias Observation of DCT Coefficients

Daichi Takeshita[1], Minoru Kuribayashi, Nobuo Funabiki
Okayama University, Okayama, Japan
E-mail: [1] pryf48fy@s.okayama-u.ac.jp

***Abstract*—Photographs taken by smartphones and camera devices are generally compressed using JPEG by default when they are saved. If such an image is edited, it is decompressed and processed, and then recompressed through JPEG. Therefore, an edited image must be compressed by JPEG more than once. Using this characteristic, a forensic technique has been studied to detect image tampering by detecting distortions caused by double compression. In our previous study, to analyze the JPEG compression history using a convolutional neural network and (CNN), we observed a histogram calculated from the low-frequency components in 8×8 sized blocks of images having a pixel resolution of 512×512. However, there have been no detailed considerations regarding the range of observed histograms or the selection of DCT coefficients used to extract the features from a given image. In this study, we first examine the range of histograms to measure the usefulness of the classification of double JPEG-compressed images, and then examine the classification accuracy by increasing the number of DCT coefficients observed in the low-to mid-frequency components. Our experiment results indicate that $[-40, 40]$ is an appropriate range for observing a histogram, and the selection of DCT coefficients strongly depends on the image size because of the difference in the amount of useful statistical information available.**

I. Introduction

Owing to the recent advancements made in computers and smartphones, image editing has become an easy and popular task. However, this has also caused a new problem known as image forensics, in which a malicious party can intentionally modify images to mislead other reviewers. It is therefore necessary to investigate forensic techniques when analyzing for traces of image modification.

The use of a fragile watermark is one of the methods applied to detect processed images. If an embedded watermark is broken, the image is considered to have been processed. However, this technique requires the watermark information to be embedded into the image beforehand. It is therefore difficult to detect images processed through techniques that are in wide use today.

In general, images taken with digital cameras and smartphones are compressed within the device using JPEG. When a JPEG image is processed, the compressed data are first decompressed and then edited. The image is then recompressed. Therefore, an edited image must be compressed more than once. Owing to the nonlinear and lossy operations applied by a JPEG compression algorithm, the distortion characteristics of single and double JPEG compressed images differ. By analyzing the traces in a given image, the double JPEG compression detection method can classify whether the image has been compressed twice [1]. The objective of this research is to classify a given JPEG image that has been compressed one or more times.

In a previous study [2], an image with a pixel resolution of 512×512 was converted from RGB into YCbCr components, as used in JPEG compression, where each component was divided into 8×8 sized blocks, the Y component was transformed using a discrete cosine, and a histogram was created using specific DCT coefficients and trained through a CNN [3]. We detected compression twice using this process. To detect the processing of a portion of an image, it is necessary to detect a smaller image. The smaller the image applied, the more difficult it is to detect the use of a conversion process. However, we did not check how much the accuracy will decrease. In addition, we did not check the range of histograms that should be observed or which DCT coefficients should be used to create histograms for achieving a higher accuracy.

In this study, we first varied the range of the histogram to determine the change in accuracy. Next, to reduce the loss of accuracy, we experimentally adjusted the parameters and verified the components that are beneficial to the process. We then checked for the presence of parameters that depend on the image size and other parameters. Finally, to the training data, we added images that were processed by an attacker to reduce the traces of image processing. We hypothesized that this method will be able to detect such images. We also assumed that the attacker would use a median filter and tested this hypothesis experimentally.

II. Preliminaries

A. JPEG Algorithm

JPEG compression considers two features of the human eye. First, the human eye can recognize the low-frequency components of the image brightness in detail, whereas it is insensitive to the high-frequency components. Second, it has the ability to recognize the brightness of an image in detail, while being insensitive to changes in color. Using these two characteristics, the data are compressed by eliminating the high-frequency regions and color-difference components of the image. The JPEG compression process is shown in Fig. 1. First, the RGB components of the color image are

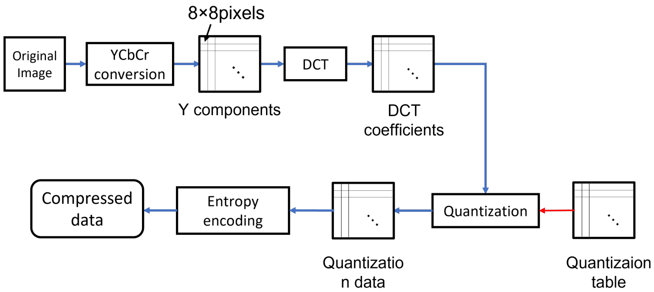

Fig. 1. JPEG algorithm.

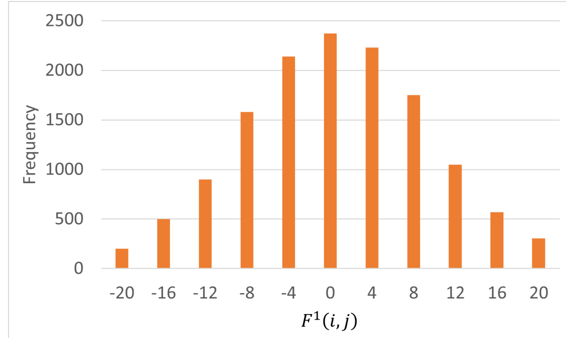

Fig. 2. Histogram of $F^1(i,j)$ compressed using $Q_{QF_1}(i,j) = 4$.

converted into YCbCr components, and each component is divided into blocks with a pixel resolution of 8×8. Second, the CbCr component is reduced according to the settings, and the Y component is subjected to a discrete cosine transform (DCT). Third, the data are quantized using a quantization table. Finally, entropy coding is applied to create compressed data. [4], [5]

B. Histogram of DCT coefficients

In this study, we used the histogram of the DCT coefficients in an image compressed using JPEG for detecting traces of double compression. This section describes the creation of a histogram. First, the RGB components of an image are converted into YCbCr components. Next, the luminance component is divided into non-overlapped blocks with a pixel resolution of 8×8, and each block is then transformed using DCT. The DCT coefficients are converted into integers through a rounding-off operation, whereas each DCT coefficient is quantized using the integer step size specified in the JPEG algorithm. To create a histogram, we focus on one coordinate in the 8×8 sized blocks and count the occurrence of the DCT coefficient values. For example, when focusing on the coordinate (i, j) of the DCT coefficients, a frequency of -8.3 becomes -8 when rounded into an integer. The histogram of the DCT coefficients in an image is symmetric with zero mean and smaller values appearing toward the outside.

C. Distortions of Double Compression

We describe how the DCT coefficients change during the double-compression process. Once an image is compressed by JPEG, each DCT coefficient $F(i, j)$ in each block is quantized by the quantization step $Q_{QF}(i, j)$, which is specified by a standardized quantization table and a given quality factor (QF). After the first compression with QF_1, the quantized DCT coefficient $F^1(i, j)$ becomes a multiple value of the quantization step $Q_{QF_1}(i, j)$. When the second compression is applied with $QF_2 \neq QF_1$, the quantization step $Q_{QF_2}(i, j)$ is different from the first, $Q_{QF_1}(i, j)$, which causes a change in the distribution of the histogram.

For instance, the histogram of a single JPEG compressed image is shown in Fig. 2, and that of a double JPEG compressed image is shown in Fig. 3, where $Q_{QF_1}(i, j) = 4$ and $Q_{QF_2}(i, j) = 3$. The occurrence becomes zero at $F^2(i, j) = 6$, as shown in Fig. 3. This is because $4, 8$ divided by 3 equals $1.33\cdots, 2.66\cdots$, rounded off to 1,3, and when multiplied by 3 equals $3, 9$; therefore $F^1(i, j) = 4$ quantized by 3 equals $F^2(i, j) = 3$, and $F^1(i, j) = 8$ quantized by 3 equals $F^2(i, j) = 9$. Thus, if the relationship $QF_1 < QF_2$ holds, the histogram contains zero values and distortion occurs.

Conversely, if the relationship $QF_1 > QF_2$ holds, the histogram does not contain zero values. However, the histogram does contain some unnaturally large values.

For instance, consider the case where $Q_{QF_1}(i, j) = 3$ and $Q_{QF_2}(i, j) = 4$. $F^1(i, j) = 6$ and $F^1(i, j) = 9$ at the first quantization are divided by 4 at the second quantization, so they become 1.5 and 2.25, and both are rounded to 2, and miltiplied by 4 equals 8. As a result, the value of $F^2(i, j) = 8$ in the histogram after second quantizations becomes very large, which is a distortion of the histogram.

In addition, when the JPEG images are decompressed, rounding to the nearest integer is applied. One of the steps in the decompression process is to convert the DCT coefficients into RGB. First, the DCT coefficients are converted into YCbCr components through the inverse discrete cosine transform. The YCbCr components are then converted into RGB components. At this time, the RGB component contains a small number of pixels. However, the RGB components are represented by integers from 0 to 255, and therefore need to be converted into integers. If the relationship $QF_1 = QF_2$ holds, the histogram is not distorted by quantization, but the histogram is distorted by integer rounding.

Thus, depending on the relationship between the first and second QF, the histogram will be distorted. By capturing this distortion as a feature, the CNN is trained.

When a JPEG image is edited, the file is decompressed into an RGB color space by rounding the pixel values into integers, and is recompressed through JPEG after a modification of the image. Hence, the distortions include the above-mentioned changes, as well as the rounding errors during decompression. Owing to the presence of nonlinear artifacts, it is possible to detect traces of double JPEG compression.

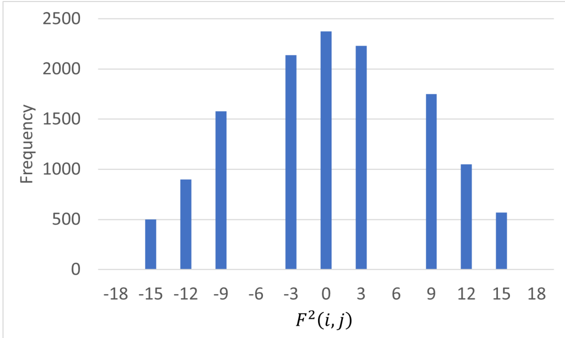

Fig. 3. Histogram of $F^2(i,j)$ compressed using $Q_{QF_1}(i,j) = 4$ and $Q_{QF_2}(i,j) = 3$.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	96	112	100	103	99

Fig. 5. Standardized quantization table and the selected positions for analysis using a conventional method [2].

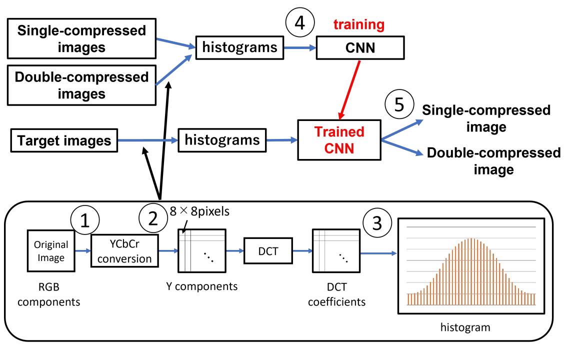

Fig. 4. The method for detecting double compression using a CNN and a histogram of DCT coefficients.

III. Conventional Method

In [2], this study assumes that the image is saved in JPEG as the original and that the image is also saved in JPEG when it is processed.

A. CNN-based Analysis

In [2], a method for detecting double compression using a CNN and a histogram of DCT coefficients of JPEG images was proposed. The process flow image is shown in Fig. 2 and described in the following section.

1) The RGB components of the training image are converted into YCbCr components.
2) The Y component is divided into blocks with a pixel resolution of 8×8, and a discrete cosine transform is applied for each block.
3) A histogram is created from the DCT coefficients obtained using the DCT coefficients in the position corresponding to a specific quantization step [6].
4) A CNN is trained using a one-dimensional vector of serially concatenated histograms.
5) Whether the compression occurred once or twice is classified using the trained CNN.

In [2], 30,000 single- and double-compressed JPEG images were used during the experiment. The pixel resolution of the image was 512×512. Five types of histograms of DCT coefficients are produced from the observation of the values within the range of $[-50, 50]$ at a quantization step size of {12,14,16,22,24}. As shown in Fig. 5, there are two coordinates for a step size of {12,22} and three for {14,16,24}, which are located among the low-frequency components. Owing to the use of multiple coordinates, statistical analyses of the quantized DCT coefficient can be conducted more accurately than the others. For example, in an image with a pixel resolution of 512×512, there are 4096 8×8 pixel-sized blocks, and the total numbers of DCT coefficients corresponding to the quantization step sizes {12,22} and {14,16,24} are 8192 and 12288, respectively.

Each histogram obtained above is normalized and concatenated as a single vector before being input into a CNN-based classifier. Because the histogram is created from DCT coefficients within the range of $[-50, 50]$, the length of each histogram is 101, and the total length of the vector is 505. The CNN-based classifier is composed of eight layers, as shown in Fig. 6. It takes a one-dimensional vector as input, applies a convolution and maximum pooling twice alternately, and then passes through two fully connected layers during the output. The convolution layer uses 100 different filters. The filter size is 3×1, and the stride, the interval used to slide the filters, is 1. The pooling layer applies maximum value pooling with a filter size of 3×1 and a stride of 2. The data obtained are converted into 1D, passed through a fully connected layer, and output as a zero or 1 in the output layer. A value of zero indicates a single compression, whereas a value of 1 indicates dual compression.

B. Problem

The classification accuracy of the method in [2] was evaluated using only images with a pixel resolution of 512×512. To detect the manipulated parts in an image, it is necessary to accommodate smaller images. It is clear that the classification accuracy decreases as the image size decreases because the total number of DCT coefficients used for the histogram decreases.

To maximize the performance, some parameters of the previous method should be managed. For instance, histograms

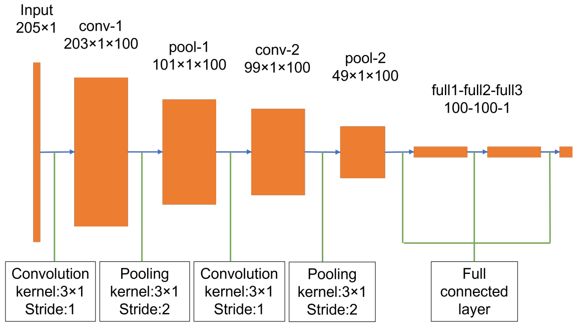

Fig. 6. Structure of the CNN used in the previous study.

were created from the values of the DCT coefficients within a range of $[-50, 50]$, despite the validity of the range not having been evaluated.

Some quantization steps in the low-frequency component appear multiple times. Therefore, the histogram is created from the DCT coefficients corresponding to the positions of $\{12,14,16,22,24\}$ in the quantization table. In terms of the accuracy of each histogram, the selection of such positions is better than that of the other coefficients. However, the other coefficients may retain useful information for the classification tasks. It has yet to be verified whether the selection of the other DCT coefficients contributes to an increase in the classification accuracy.

Therefore, in this study, we conducted experiments for the following three points, i.e., changes in the classification accuracy when the image size is reduced, when the range of the histogram is changed, and when increasing the DCT coefficients used in the histogram.

IV. Proposed Method

It is clear that the classification accuracy decreases when conventional methods are applied to smaller images. The objective of this study is to maximize the classification accuracy with respect to the following two points. One is the range of the histogram, and the other is the selection of the frequency components. We also considered supervised datasets for training the CNN-based classifier.

A. Range of Histogram

In the conventional methods mentioned in Section III, each histogram is produced from the values of the DCT coefficients within the range of $[-50, 50]$. The wider the range of the histogram, the more information that is input into the CNN-based classifier. It is therefore expected that, the wider the range of the histogram, the higher the accuracy. In a previous study, because the range of the histogram was fixed, the dependency of the classification accuracy was not investigated with respect to the difference within this range. It is necessary to verify the optimal range of the histogram and the image size. Therefore, the range of the histogram was changed to verify the change in the classification accuracy.

B. Frequency Components

Because of a lower visual importance, high-frequency components are divided by larger step sizes during quantization to reduce the number of data used in the JPEG compression. Therefore, it is difficult to extract useful information from the histogram of high-frequency components because most of the values are zero. Thus, as a suitable approach, the DCT coefficients can be selected from the low-and middle-frequency components when considering the values of the quantization step size.

In addition to the above generic characteristics of DCT coefficients, the targeted frequency components are selected for the following reasons when applying a conventional method. As shown in Fig. 5, the step size at those frequency components appears more than once, and hence, the number of samples for a statistical analysis is larger than that of the other components, which appear only once.

However, as a general case, the more information that is input into the CNN, the higher the accuracy. Hence, to collect as much useful information as possible, the other frequency components should be included in the input. This can be easily realized by increasing the number of histograms for more DCT coefficients from among the low-and middle-frequency components.

Here, we need to consider the difference in the number of samples collected from the DCT coefficients. For instance, the step sizes at the colored positions in Fig. 5 appear two or three times, whereas the others appear only once. The difference in the samples at each histogram causes a difference in the reliability of the data accuracy for the classification task. Owing to the use of a CNN-based classifier, the difference in reliability reflects the weight values in a neural network. It is therefore sufficient to train a CNN-based classifier through a feature vector concatenated using histograms.

Excluding the colored positions in Fig. 5, we select six additional positions from the low-frequency components with step sizes of $\{10,\ 11,\ 13,\ 17,\ 18,\ 19\}$, and calculate the corresponding six histograms from the DCT coefficients at those positions. Furthermore, from the middle-frequency components, six positions with step sizes of $\{26,\ 29,\ 35,\ 37,\ 40,\ 49\}$ are selected to calculate the histograms. For convenience, the original five histograms are called the R1 region, the additional six histograms from the low-frequency components are the R2 region, and six histograms from the middle-frequency components are the R3 region. Figure 7 illustrates those three regions with different colors: red is R1, orange is R2, and blue is R3.

C. Supervised Datasets

The proposed method can be used to detect the double compression of JPEG images with high accuracy. However, an attacker will modify an image to avoid leaving traces of a modification or double compression. A possible flow of such modification would be as follows. First, an original image is taken by a camera device and is automatically compressed in the device using JPEG. An attacker then decompresses the

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	96	112	100	103	99

Fig. 7. Three regions of the selected positions. Red is R1, orange is R2, and blue is R3.

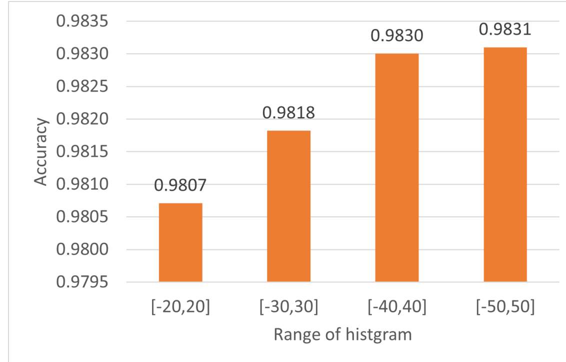

Fig. 8. Comparison of accuracy with respect to the range of the histogram.

image and modifies it in some way. Finally, the modified image is recompressed using JPEG. Namely, between the first and second compressions, the attacker will be able to conduct a filtering operation to reduce the traces of double compression. This operation degrades the classification accuracy of the proposed method. In this study, we assume that the attacker applies a median filter.

The traces of double compression will still be involved in such a modified image. Therefore, in this study, we hypothesized that it is possible to detect the traces if the modified images as well as the images created only through all double compression are used for training the proposed CNN-based classifier. To test this hypothesis, we created a training model with and without the modified images and evaluated the classification accuracy.

V. Experimental Results

A. Environment

The PC environment and software versions for the simulations conducted in this study are listed in Table I.

TABLE I
PC environment used for the simulations.

CPU	AMD Ryzen 7 3800X
RAM	64GB (DDR4-3200)
GPU	Nvidia GeForce RTX2080 SUPER (8GB)
OS	CentOS 8.3.2011
Python	3.7.6
tensorflow	2.1.0
keras	2.3.1

In total, 30,000 single-compressed images and 30,000 double-compressed images were used in the experiment. All images were taken from [7]. The ratio of training to validation was 4:1, i.e., 24,000 images were used for training and 6000 images were applied for validation. A total of 30,000 images were used for the testing. The numbers of QFs used for compressing these images were randomly selected from among the following set: {70, 75, 80, 85, 90, 95}.

We prepared 5 image sizes, i.e., $N \times N$: $N = \{512$, 256, 128, 64, 32$\}$, and conducted a total of 15 experiments. The image was resized by cropping it at the center of the original image. In this way, we examined how the accuracy changes depending on the number of histogram types, that is, the number of DCT coefficients used in the histogram. We also examined how these differences affect the results when the image size is reduced.

In this study, we used the "accuracy" as an evaluation index of CNN training and "binary cross entropy" as a loss function. Here, "accuracy" refers to the percentage of correct answers, which is the probability that the trained model will be able to correctly distinguish whether the image has been compressed one or more times. The CNN model is trained by iteratively applying training and validation processes at each epoch. During training, the model is updated by calculating the "accuracy" and "loss" on the training data. Subsequently, the trained model is validated on data that differ from the training data. The iteration of the processes ends when the validation loss does not increase for 15 consecutive epochs. Each experiment was conducted 10 times, and the average accuracy was calculated.

B. Classification of Double Compression

First, the range of the histogram $[-T, T]$ is changed into the following four types: $T = \{20, 30, 40, 50\}$. The accuracy is shown in Fig.8. The classification accuracy improved as the range expanded. However, the accuracy does not improve proportionally with the same slope, and the slope decreases within the range from $[-40, 40]$ to $[-50, 50]$, suggesting a small improvement in accuracy within a range of larger than $[-40, 40]$.

The number of epochs applied is shown in Fig. 9. As shown in Fig. 9, the larger the range of the histogram is, the smaller the number of epochs. This is because the larger the range of the histogram is, the greater the number of data, which makes it easier to capture the features of modifications and to finish the learning process quickly. However, there is not much difference between epoch numbers of $[-40, 40]$ and $[-50, 50]$. Therefore, the time required for learning is not expected to

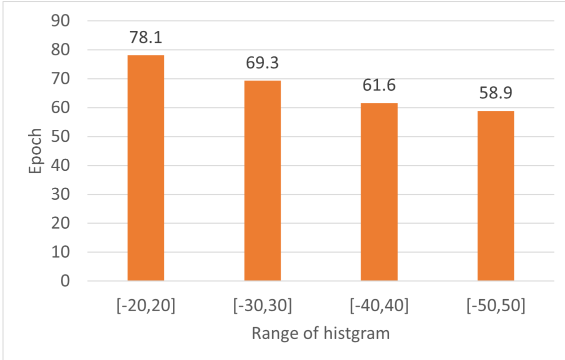

Fig. 9. Comparison of the number of epochs with respect to the range of the histogram.

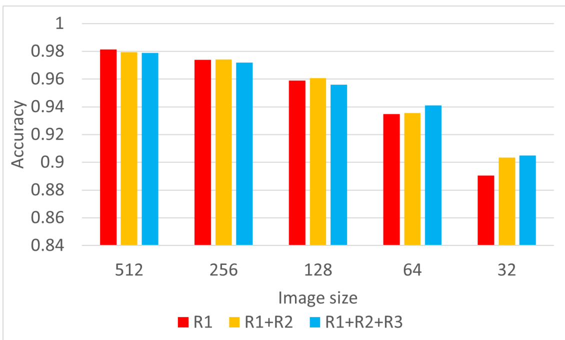

Fig. 10. Comparison of accuracy between different types of histogram and different image sizes.

dec'rease even if the range is wider. The larger the range of the histogram is, the longer it takes to create. For these reasons, based on the results under our experimental conditions, we can state that the range of [-40, 40] is suitable.

Next, within a histogram range of $[-40, 40]$, we evaluate the classification accuracy for five image sizes $N \times N$ of $N = 512, 256, 128, 64, 32\}$ and three regions of frequency components ({R1, R1+R2, R1+R2+R3}). The simulation results are presented in Fig. 10.

As expected, the accuracy decreases with a decrease in the image size because the amount of information input into the CNN decreases. In the simulation results shown in Fig. 10, for $N \geq 128$, the classification accuracy is higher with fewer types of histograms, and for $N \leq 64$, the classification accuracy is higher with a larger number of histogram types. First, for $N \geq 128$, the results are contrary to the expectation that increasing the number of types of histograms can increase the amount of information input to the CNN and improve the accuracy. When the image size is large, R1 is sufficient for learning, whereas R1+R2 or R1+R2+R3 may hinder the classification. In addition, for $N \leq 64$, the number of pixels is small and the amount of information is less than that of a large image size, thus increasing the number of histograms to increase the amount of information is considered to increase the accuracy.

C. Median Filtering Forensics

First, we examined the classification accuracy of the median-filtered images using the trained model described in Section V-B. In this experiment, we used median-filtered images as a double-compressed image of the evaluation data. This experiment assumes that the attacker applies a median filter between the first and second compressions to reduce the traces of double compression. The simulation results are presented in Fig. 11. As expected, a considerable decrease in accuracy was observed in the results. Overall, the accuracy decreases by approximately 20% from the results detailed in Section V-B. The results also show that the loss was quite high, ranging from 5 to 20 in all cases, whereas the loss was approximately 0.1 with the approach described in Section V-B. The loss is measured along with the accuracy when evaluating the function shown in Section V-A. It becomes larger when the inferred value and the correct answer are far apart. In other words, the smaller the loss is, the better the classification accuracy. If the loss is between 5 and 20, the model does not function correctly.

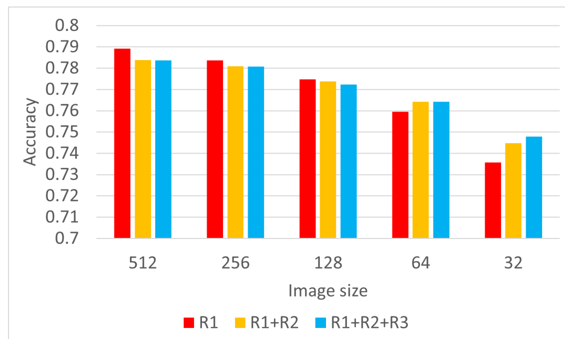

Fig. 11. Accuracy of evaluations involving median-filtered images using the trained model described in Section V-B.

Next, we added median-filtered images to the training data described in Section V-B. Subsequently, the same evaluation data as in the previous experiment were used for the evaluation. The training data consisted of 30,000 single-compressed images and 15,000 double-compressed images, as indicated in Section V-B, along with 15,000 double-compressed images with median filtering. The simulation results are presented in Fig. 12. The classification accuracy is much higher than that described in Section V-B. The experiment results show that the accuracy in the case of R1+R2+R3 is larger than that of the other cases. This indicates that the use of more DCT coefficients is effective for images with reduced traces of modifications.

By adding the training data with median-filtered images, the accuracy of the classification can be improved, thereby confirming the validity of our hypothesis. Even if the traces of double compression are reduced through the filtering operation, the CNN-based classifier still classifies the double-compressed image by predicting the operation applied by the attacker.

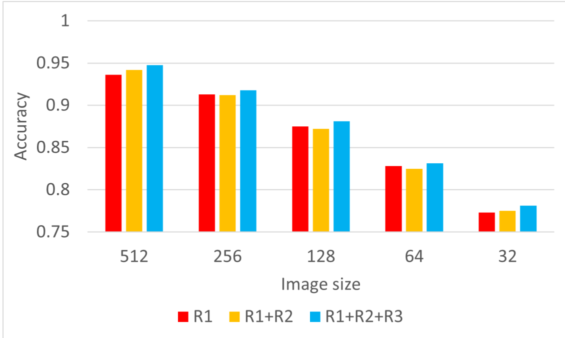

Fig. 12. Accuracy of evaluations involving median-filtered images using the trained model applied in this experiment.

We also evaluated the classification accuracy of the non-filtered images for the CNN-based classifier trained on the median-filtered images. The simulation results are presented in Fig. 13. This indicates that the classification is highly accurate. However, the accuracy is slightly lower than that of the classifier trained using images with median filtering. This indicates that applying only the learning model obtained through this study will not provide the best accuracy in the detection of modifications under any cases.

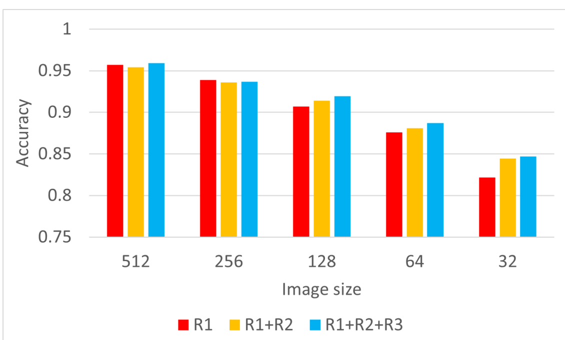

Fig. 13. Accuracy of evaluations of non-median-filtered images using the trained model.

A possible solution to this problem is to combine the two learning models obtained through this study with that described in Section V-B. For example, it is possible to create a learning model with better accuracy by applying ensemble learning. The classification of median-filtered images using the learning model described in Section V-B results in a fairly high loss. We expect this feature to be successfully applied to ensemble learning.

In addition, if we know in advance whether the target image has been median filtered, we can select a model with high accuracy for evaluation. Therefore, we can consider a method for finding modifications after detecting traces of the median filter, as studied in [8].

VI. Conclusions

In this study, we evaluated the range of histogram and DCT coefficients that are effective for the analysis of a double compression detection method when applying a histogram with a CNN, as described in a previous study. Using the optimal range of the histogram obtained from the results, the number of DCT coefficients observed in the low-to-mid frequency components was increased to investigate the classification accuracy of the JPEG compression history. As a result, the range of the histogram was considered to be appropriate at $[-40, 40]$. In the simulations applying this range, the accuracy was higher with fewer types of histograms for larger sized images, and higher with more types of histograms for smaller sized images.

We also conducted experiments under the assumption that a median filter was applied to reduce the traces of double compression. The classification accuracy can be improved by adding images that have been preprocessed using the assumed operations.

Acknowledgment

This research was supported by JSPS KAKENHI Grant Number 19K22846, JST SICORP Grant Number JPMJSC20C3, and JST CREST Grant Number JPMJCR20D3, Japan.

We would like to thank Editage (www.editage.com) for English language editing.

References

[1] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proc. Digit. Forensic Res. Workshop (DFRWS)*, 2003, pp. 5–8.

[2] K. Oura, M. Kuribayashi, and N. Funabiki, "CNN-based analysis on editing history of JPEG file using histogram of DCT coefficients," in *Technical Report of IEICE*, 2020, vol. 119, pp. 91–96.

[3] K. Saito, *Deep Learning from the Basics*, Packt Publishing, 2021.

[4] Alain M Leger, Takao Omachi, and Gregory K Wallace, "Jpeg still picture compression algorithm," *Optical Engineering*, vol. 30, no. 7, pp. 947–954, 1991.

[5] S.Chandra and C.Schlatter Ellis, "Jpeg compression metric as a quality-aware image transcoding," in *UNIX Symposium on Internet Technologies Systems*, 1999.

[6] R.Zhang Q.Wang, "Double JPEG compression forensics based on a convolutional neural network," in *EURASIP Journal on Information Security*, 2016, number 23.

[7] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato, "Raise: A raw images dataset for digital image forensics," in *Proceedings of the 6th ACM multimedia systems conference*, 2015, pp. 219–224.

[8] J.Miao G.Duan and T.Huang, "Median filtering detection of small-size image using alexcaps-network," in *Proc.International Workshop on Digital-forensics and Watermarking (IWDW2019)*, 2019, vol. LNCS 120022, pp. 126–140.