DHNet: Double MPEG-4 Compression Detection via Multiple DCT Histograms

Seung-Hun Nam
NAVER WEBTOON AI

Jihyeon Kang
NAVER WEBTOON AI

Wonhyuk Ahn
NAVER WEBTOON AI

In-Jae Yu
Samsung Electronics

Myung-Joon Kwon
Korea Advanced Institute of Science and Technology

Abstract—In this article, we aim to detect the double compression of MPEG-4, a universal video codec that is built into surveillance systems and shooting devices. Double compression is accompanied by various types of video manipulation, and its traces can be exploited to determine whether a video is a forgery. To this end, we present a neural network-based approach with discriminant features for capturing peculiar artifacts in the discrete cosine transform (DCT) domain caused by double MPEG-4 compression. By analyzing the intra-coding process of MPEG-4, which performs block-DCT-based quantization, we exploit multiple DCT histograms as features to focus on the statistical properties of DCT coefficients on multiresolution blocks. Furthermore, we improve detection performance using a vectorized feature of the quantization table on dense layers as auxiliary information. Compared with neural network-based approaches suitable for exploring subtle manipulations, the experimental results reveal that this work achieves high performance.

Index Terms: Video forensics, Double MPEG-4 compression detection, Multiple DCT histograms, Auxiliary feature, Compression artifacts

Multimedia forensics is a field of research aimed at verifying the integrity of multimedia content and detecting fine-grained artifacts caused by multimedia forgery. Due to the rapid proliferation of shooting devices and user-friendly editing software (e.g., Adobe Photoshop and Premiere Pro), the volume of forged content is increasing, and multimedia forensic research is becoming ever more significant. Usually, manipulations are performed so exquisitely that their traces are difficult to distinguish via the human visual system (HVS); thus, forensic techniques have been designed to explore and capture subtle changes. A fundamental assumption underlying multimedia forensics is that intrinsic statistical fingerprints of pristine content, such as acquisition and compression artifacts, are different from forged ones. Noting this point of view, forensic researchers have developed techniques on various...
media, including image, video, and audio.

Recently, under the influence of the advances in video-sharing platforms, research on video forensics has been conducted, with the following various targets: double compression, frame-rate conversion, and inter-frame forgery. As illustrated in Figure 1, double compression is a weighty indicator of video forgery. Due to the storage issue, uncompressed videos are generally single-compressed before distribution, and when videos have been tampered with by an editing tool on the malicious distributor side, a second compression occurs during the storage process. As the video forgery process commonly involves double compression, detecting recompression traces helps verify the integrity of a video.

Focusing on this issue, many researchers aimed to detect double compression in video and chose MPEG-4 part 2 (hereinafter referred to as MPEG-4) as the target codec, a typical lossy compression standard developed by the MPEG group and widely adopted in surveillance systems, shooting devices, and editing tools. MPEG-4 is largely composed of intra and predictive coding, which are used to reduce spatial and temporal redundancy, respectively. In particular, intra coding is applied to the intra-coded frame (I-frame) located in the first of each group of pictures (GOP), where quantization is performed on the discrete cosine transform (DCT) domain of each divided 8 × 8 block. When considering double compression, if the first and second quantization tables have differences, the distribution of the corresponding DCT coefficients differs from that of the single-compressed media.

In the past, approaches with handcrafted features (e.g., Markov statistics-based features) were dominant to distinguish the minute statistical variations on the decoded I-frame. After the superior performance of the data-driven approach using a convolutional neural network (CNN) was demonstrated, network architecture, allocating preprocessing layer or unpooled layers in the initial layers to capture low-level signals, was proposed. Recently, approaches using both the CNN and handcrafted features (e.g., DCT histogram and vectorized quantization matrix) demonstrated excellent performance in detecting double compression in JPEG files. Since intra coding is based on a JPEG-like scheme, it is crucial to investigate the efficacy of these features in terms of detecting double MPEG-4 compression.

Motivated by the mentioned issue, neural network-based video forensics (referred to as DHNet) using discriminant features is presented. To explore and learn distinguishable footprints caused by block-DCT-based double quantization, we devise multiple DCT histograms and auxiliary features suitable for capturing subtle statistical characteristics of the DCT coefficients. Extensive experiments with large datasets demonstrated that our method outperformed comparable methods. The main contributions are as follows:

- To the best of our knowledge, this is the first attempt to employ multiple DCT histograms, which are designed to focus on the statistical properties of the DCT coefficients of multi-resolution blocks as features for detecting double MPEG-4 compression.
- By analyzing the differences between MPEG-4 and JPEG in terms of quantization table acquisition, we propose auxiliary features with the base quantization table and quantizer scale, which contribute to the performance improvement of DHNet.

The remainder of this paper is organized as follows: In “Background” section, we review the intra coding of MPEG-4, cases of double MPEG-4 compression, and the previous works related to this work. Then, “Proposed Method” section introduces the proposed forensic approach utilizing distinctive features, followed by the performance analysis in “Experiments” section. Finally, the article is concluded in “Conclusion” section.
Figure 2. Block diagram of MPEG-4 and examples of the visualization of the residual between decoded single- and double-compressed I-frames, where \( q_{s1} = 3 \) and \( q_{s2} = 5 \).

**BACKGROUND**

Analysis of Intra Coding in MPEG-4

For the MPEG-4 standard, the GOP is exploited as the base encoding unit, and each GOP comprises an I-frame, predictive frame (P-frame), and bidirectionally predictive frame (B-frame). The I-frames are coded using intra coding based on DCT and quantization to reduce spatial redundancy (see Figure 2(a)). The P-frames are predicted from previous I- or P-frames exploiting motion estimation and compensation to reduce temporal redundancy. In this work, B-frames encoded from past and future frames are not considered for simplicity, and compression quality is controlled in variable bitrate (VBR) mode for efficient bitrate allocation. In VBR mode, quality is determined by the quantizer scale \( q_s \), and a value closer to 1 indicates the highest level.

Examples of compression artifacts remaining in decoded I-frame, caused by double compression, are illustrated in Figures 2(b)–2(h). First, Figures 2(b) and 2(e) show examples of decoded I-frames of single- and double-compressed videos with the configuration of \((q_{s1}, -)\) and \((q_{s1}, q_{s2})\), respectively. No significant difference was observed by the HVS; however, when observing partially textured and flat areas of enlarged images (Figures 2(c) and 2(f)), subtle differences can be identified. To focus on these fine distortions, we visualized the residual image between the y-channel of the enlarged images and observed perceptually distinct artifacts (see Figure 2(h)). These footprints, distinguishable from single compression, occur in rounding errors that are inevitably caused in the double quantization of intra coding.

In detail, intra coding divides a frame into 8 × 8 blocks and then quantizes the DCT coefficients of each block with \( q_s \) and the default quantization matrix \( Q_m \in \mathbb{N}^{8 \times 8} \). We let \( C(i, j) \) and \( C_q(i, j) \) be block-DCT coefficients and quantized coefficients at coordinates \((i, j)\), respectively. As depicted in work \(^6\), the quantization process is as follows:

\[
C_q(i, j) = (C(i, j) \times S(Q_m(i, j), q_s) + r) \gg (s_b - 3),
\]

where \( S(\cdot) \), \( r \gg x \), and \( s_b \) denote the function to generate a scaled quantization matrix, the rounding value, a right-shift by \( x \) bits on binary numbers, and the default scale-bits, respectively. The dequantized coefficients \( C_d(i, j) \) can be generated from the dequantization process as follows

\[
C_d(i, j) = (C_q(i, j) \times Q_m(i, j) \times q_s) \gg 3.
\]

Unlike JPEG, the elements of the fixed quantiza-
tion table in intra coding are prescaled due to $q_s$; hence, the divisor relationship between $q_s$ and $q_{s2}$ is an important issue for tracing the rounding error caused by double quantization. This article excludes the case in which $q_{s1} = q_{s2}$ in consideration of this constraint, which is a more realistic condition. By analyzing intra coding, we identified the following important points.

- First, as intra coding employs JPEG-like approaches, we were inspired to adopt a DCT histogram suitable for detecting double JPEG. In addition, multiple histograms of multiresolution blocks may help solve a given task in that they comprehensively explore the statistical properties of the DCT domain.

- Secondly, $q_s$ and $Q_{cm}$ are essential parameters in intra coding; hence, they can be usefully exploited as auxiliary features to help a CNN learn double compression artifacts.

Case Study of Double Compression

In VBR mode, $q_s$ and GOP size $g$ are key configurations of MPEG-4. As presented in Figure 3, the following double compression cases occur according to their combination: aligned and non-aligned GOP structures. First, for the aligned GOP structure ($g_1 = g_2$), quantization-based intra coding is periodically applied to the first frame of each GOP; hence, double quantization with a different quantization scale inevitably causes rounding errors, leaving distinguishable artifacts against single compression (see the dashed red and blue boxes in Figure 3). Next, for the non-aligned GOP structure ($g_1 \neq g_2$), inferring the index of the double-compressed I-frame in a given video except for the first frame is a challenging task. To handle this issue, we trained a model with datasets of massive decoded I-frames generated for the aligned case and then aimed to infer the first I-frame of the non-aligned case to the pretrained model. The countermeasures for non-aligned cases are detailed covered in the experimental section.

Related Work

Conventional approaches for detecting double compression were based on handcrafted features and statistical characterization to explore fingerprints caused by the complicated processes of the target codec. To capture fine artifacts, Sun et al. and He et al. employed the first digit distribution of quantized coefficients and block artifacts combined with the variation of prediction footprints, respectively. Jiang et al. presented a linear discriminant analysis-based method using the Markov feature in the DCT domain. Aghamaleki and Behrad employed feature vectors obtained from the GOP and support vector machines. Yao et al. used temporal segmentation to capture abnormal artifacts in rapid and static periods. Currently, video forensics targeting high-efficiency codec (e.g., HEVC) is being...
actively studied. Moreover, approaches based on the distribution of the DCT domain in a transform unit\textsuperscript{12}, analysis of the quality degradation mechanism in intra coding\textsuperscript{13}, and analysis of the quality degradation process in inter coding\textsuperscript{14} were presented.

With the advances in CNNs, Boroumand et al.\textsuperscript{7}, Nam et al.\textsuperscript{15}, and Yoon et al.\textsuperscript{16} demonstrated that placing an unpooled layer on the initial layer helps capture low-level signals (e.g., stego signals, local artifacts, and interpolation artifacts). With the help of the preprocessing with a high-pass filter, He et al.\textsuperscript{2} and Nam et al.\textsuperscript{3} proposed CNNs to classify relocated I-frames and double-compressed I-frames, respectively. To detect double JPEG compression accompanying an $8 \times 8$ unit quantization, such as intra coding in MPEG-4, Barni et al. used a CNN-based approach exploiting the DCT histogram of all DC and AC components\textsuperscript{5}. Park et al. improved upon the method\textsuperscript{5} by providing a quantization table in the header to the CNN, which exhibited performance improvement\textsuperscript{8}. Kwon et al. presented a JPEG artifact learning module (J-ALM) using the DCT volume representation and quantization matrix to learn the compression artifacts\textsuperscript{17}.

Inspired by the previous works, we devise discriminant features to capture peculiar artifacts and design multi-stream architecture that effectively learns those features.

**PROPOSED METHOD**

Figure 4 presents the whole procedure of the proposed approach consisting of the following three steps: multiscale DCT histogram generation, auxiliary feature generation, and multistream CNN-based feature learning.

**Preliminaries**

Let video $v$ with resolution $W \times H$ be represented by $v = \{ ..., f_t, ... \}$, where $f_t$ denotes a $t$-th decompressed frame in RGB format. The proposed approach uses the decompressed I-frame indicated by $I = \{ f_t \} \in \mathbb{R}^{W \times H \times 3}$ as input. Here, $t_i$ denotes the index of the I-frame. The preprocessed input is defined as $I_y = \{ Y(I) \} \in \mathbb{R}^{W \times H \times 1}$, where $Y(\cdot)$ denotes the Y-channel of the input after RGB-to-YUV conversion.

**Multiple DCT Histogram Generation**

Inspired by an approach\textsuperscript{5} using a single histogram to detect double JPEG, we devised multiple DCT histograms as features to focus on the statistical properties of DCT coefficients on multiresolution blocks of sizes $4 \times 4$, $8 \times 8$, and $16 \times 16$. For simplicity, the size of each block is defined as $\delta \times \delta$, where $\delta = \{ 4, 8, 16 \}$, and the generated DCT histogram for each block size is denoted as $F_{h_{\delta}}$. This subsection focuses on the $F_{h_{\delta}}$ generation process, and the detailed procedure is illustrated in Figure 4. First, the
DCT coefficients of each $\delta \times \delta$ block of $I_g$ are computed. For frequency-specific analysis, the obtained coefficients are reshaped into $\hat{D}_\delta$ (size of $W/\delta \times H/\delta \times \delta^2$) with the same frequency component for each channel. $\hat{D}_\delta$, which indicates each channel of $\hat{D}_\delta$, is the output of a 2D convolution operation with stride $\delta$ between $I_g$ and $H_c$, where $H_c$ is a 2D DCT basis for frequency $c$, $c \in \{(1, 1), (1, 2), \ldots, (\delta, \delta)\}$.

For each channel, a cumulative histogram is calculated based on $\hat{D}_c$ and the boundary value $b$ of the bin in the histogram, where $b \in \{-\alpha, -\alpha + 1, \ldots, \alpha\}$. As depicted in work\(^5\), $B_{c,b}$ is the $b$-th bin in a cumulative histogram for $c$ and it means the average number of values in $\hat{D}_c$ that are greater than $b$. $B_{c,b}$ is computed as follows:

$$B_{c,b} = \frac{\delta^2}{W \times H} \sum_{j=1}^{H/\delta} \sum_{i=1}^{W/\delta} T(\hat{D}_c(i,j) - b), \quad (3)$$

where $T(\cdot)$ indicates the threshold function that turns positive and negative numbers into 1 and 0, respectively. Computing $B_{c,b}$ for all $c$ and $b$ completes the cumulative histogram and DCT histogram feature $F_h$ can be generated by calculating the difference between adjacent bins in the cumulative histogram: $F_{hs} = \{f_c|f_{c,b} = B_{c,b+1} - B_{c,b}, \forall c,b\}$, where the size of $F_{hs}$ is $2\alpha \times \delta^2 \times 1$. Figure 4 reveals an intuitive explanation of histogram generation, and we used multiple DCT histograms generated by following the mentioned procedures.

**Auxiliary Feature Generation**

The variables $Q_m$ and $q_s$ indicate a quantization matrix of $8 \times 8$ and the quantizer scale value of the last compression, respectively. Unlike JPEG, in which a quantization table corresponding to compression quality is stored in the header, quantization and dequantization on intra coding are conducted based on $Q_m$ and $q_s$. Inspired by previous work\(^5\), we expected that these include useful information when the CNN learns double compression artifacts; hence, we devised the auxiliary feature $F_a$ based on $Q_m$ and $q_s$. The proposed feature $F_a$ can be computed as: $F_a = V(Q_m) \times q_s$, where $V(\cdot)$ denotes a vectorization function and the size of $F_a$ is $64 \times 1$. In this work, we considered and investigated the following two types of tables:

$$Q_{m1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad Q_{m2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (4)$$

where $Q_{m1}$ is a matrix of ones which has no contextual meaning, and $Q_{m2}$ is a default quantization table on intra coding. In particular, $Q_{m1}$ and $Q_{m2}$ were expected to help the CNN in $q_s$-focused learning and quantization process inference, respectively. In this article, the CNNs employing $Q_{m1}$ and $Q_{m2}$ as auxiliary features are referred to as CNN-Q1 and CNN-Q2, respectively.

**Network Architecture**

We designed a multistream CNN using multiple DCT histograms of $F_{hs}$, $F_{hs'}$, $F_{hs''}$, and the auxiliary feature $F_a$ as input. As illustrated in Figure 4, the generated DCT histograms are input into each stream, consisting of three base blocks. Each base block consists of $3 \times 3$ or $1 \times 1$ convolutional (Conv) layers, each of which is followed by batch normalization, and a rectified linear unit (ReLU). The $3 \times 3$ Conv layer learns the relationship between neighboring elements, and the $1 \times 1$ Conv layer learns the association between sequentially placed feature maps. The last layer of each block was applied with $2 \times 2$ max pooling with a stride of 2 to reduce the dimensionality of the feature maps. After flattening the outputs of the last base block of each stream, the flattened feature was concatenated and input into the fully connected (FC) layers block, consisting of the FC layers, ReLU, and softmax. To induce DHNet to learn auxiliary features, we concatenated the generated $F_a$ with flattened features and the activations of two FC layers. The output of the last FC layer was input into a two-way softmax to predict the class label.

**Loss Function**

We define the loss function as follows: $L = L_c + L_r$, where $L_c$ and $L_r$ are the cross-entropy loss and regularization term, respectively. $L_c$ is computed as: $L_c = -(1 - l) \times \log \left( \frac{e^{y_0}}{e^{y_0} + e^{y_1}} \right) - l \times \log \left( \frac{e^{y_1}}{e^{y_0} + e^{y_1}} \right)$, where the output of the network is $y = [y_0; y_1]$ ($l = 0$ if the $I_g$ is single-compressed and $l = 1$ if the $I_g$ is double-compressed). To reduce the chance of overfitting, $L_r$ is defined as: $L_r = \gamma \sum_{n=1}^{N} \|w_n\|_2^2$, where $w_n$ represents the weight matrix of the $n$-th Conv layer.
**EXPERIMENTS**

**Experimental Settings**

**Datasets** We employed uncompressed videos from the following datasets: MCL-V\(^{18}\), XIPH1\(^{19}\), and XIPH2\(^{19}\). Examples of video samples constituting each dataset, information on the details of the videos, and configuration of compression are depicted in Figure 5. Before the first compression, we obtained 988 cropped videos with the resolution 256 × 256 using a non-overlapped crop in the spatial domain with various content. The videos were divided into three sets for training, validation, and testing (with an 8 : 1 : 1 ratio). For compression, libxvid in FFmpeg was used, and the GOP structure was set to \{IPPPPP\}. \(q_1\) and \(q_2\), which control the quality of the first and second compressions, were set to \{3, 5, 7\} in VBR mode. Based on the literature\(^6\), it was not considered double compression when \(q_1 = q_2\).

With these parameters, 2,964 (988 × 3) single-compressed videos and 5,928 (988 × 3 × 2) double-compressed videos were obtained. To train the DHNet and evaluate the performance of frame-wise detection, we obtained 120,000 decoded I-frames from the generated videos where the ratio of positive to negative samples is 1:1. The \(q_s\) and base quantization tables for auxiliary features were stored in the database along with the generated decoded frames. In the experiments on temporal and GOP-wise detection, the single- and double-compressed videos generated for testing were used. The setting for response to non-aligned GOP structure (i.e., \(g_1 \neq g_2\)) is introduced in the last subsection.

**Baselines** We compared the proposed DHNet with the following eight CNN-based methods for multimedia forensics and steganalysis: HeNet\(^2\), H-VGG\(^3\), FCDNet\(^{16}\), SRNet\(^7\), BarniNet\(^5\), ParkNet\(^8\), and J-ALM\(^{17}\). For HeNet and H-VGG, high-pass filtering (e.g., \([-1, -2, -1; 2, 6, -6, 2; -2, 8, -12, 8, -2, 6, -6, 2; -1, 2, -2, 2, -1]\)) was used as the preprocessing layer to focus on the traces remaining in a frame. For BarniNet and ParkNet, a histogram based on an 8 × 8 block-DCT was used as input. The auxiliary features were provided as optional for ParkNet, but were provided as essential for J-ALM. Furthermore, we employed MF-LDA\(^6\), a non-CNN approach using five GOPs for the performance comparison of the GOP-wise detection.

**Evaluation Metrics** Detection performance was measured using the accuracy (ACC), true negative rate (TNR), precision (PRE), recall (REC), and F1-score (F1-S). Each metric is defined as follows: ACC (%) = \(\frac{TP + TN}{TP + TN + FP + FN} \times 100\), TNR (%) = \(\frac{TN}{TN + FP} \times 100\), PRE (%) = \(\frac{TP}{TP + FP} \times 100\), REC (%) = \(\frac{TP}{TP + FN} \times 100\), and F1-S (%) = \(2 \times \frac{PRE \times REC}{PRE + REC} \times 100\), where \(TP, TN, FP,\) and \(FN\) are the numbers corresponding to the true positives, true negatives, false positives, and false negatives, respectively. In addition, the receiver operating characteristic (ROC) curves were computed to compare the performance of the proposed and comparative methods. The ROC curve is defined as a plot of the true positive rate against the false positive rate, and the area under the curve (AUC) is further used as a metric.
Learning rate of 2080 Ti. We conducted experiments using NVIDIA GeForce RTX 2080 Ti. Using the Adam optimizer with a β1 constant of 0.9 and β2 = 0.999, and the numerical stability constant ϵ = 10−8. The mini-batch size, α, and γ were set to 32, 60, and 10−4, respectively. The selected model maximized the detection accuracy on the validation set for 60 epochs.

### Results on Frame-wise Detection

First, we conducted frame-wise detection with comparative CNN-based methods, which are specialized in exploring quantization artifacts on block-DCT or detecting fine-grained signals. Table 1(a) lists the performance metrics for each model for frame-wise detection, and we computed the ROC curve for a detailed performance analysis (see Figures 6(a) and 6(b)). Without the help of auxiliary features, the ACC, TNR, PRE, REC, F1-S, and AUC of the DHNet were 88.51%, 87.78%, 87.95%, 89.23%, 88.59%, and 0.9553, respectively. Importantly, DCT histogram-based approaches, including BarniNet, ParkNet, and our DHNet, effectively detected double compression artifacts on the decoded I-frame, achieving an ACC of over 85%. Particularly, among them, DHNet exhibited the best performance for all approaches, including BarniNet, ParkNet, and DHNet.

### Implementation Details

Using the Tensor-Flow framework, we built the network and conducted experiments using NVIDIA GeForce RTX 2080 Ti. We used the Adam optimizer with a learning rate of 10−4, momentum coefficients β1 = 0.9 and β2 = 0.999, and the numerical stability constant ϵ = 10−8. The mini-batch size, α, and γ were set to 32, 60, and 10−4, respectively. The selected model maximized the detection accuracy on the validation set for 60 epochs.

### Results on GOP-wise Detection

(a) Experimental results on frame-wise detection.

(b) Experimental results on GOP-wise detection.

| Method      | ACC ↑ | TNR ↑ | PRE ↑ | REC ↑ | F1-S ↑ | AUC ↑ | ACC on single comp. (q1−s−) | AUC on double comp. (q1−s−) |
|-------------|-------|-------|-------|-------|--------|-------|-----------------------------|-----------------------------|
| DHNet       | 86.32 | 85.72 | 84.92 | 84.39 | 84.29  | 0.9536| 96.29 86.29                 | 98.82 96.65                 |
| ParkNet     | 90.57 | 88.70 | 87.86 | 87.22 | 87.15  | 0.9553| 97.14 96.14                 | 99.18 98.97                 |
| DHNet-Q1    | 92.37 | 91.98 | 91.56 | 91.14 | 90.86  | 0.9587| 99.54 99.13                 | 100 99.77                  |
| DHNet-Q2    | 92.37 | 92.98 | 92.90 | 91.76 | 92.33  | 0.9587| 99.54 99.13                 | 100 99.77                  |

Table 1. Performance comparison with comparative methods for classifying double MPEG-4 compression.

Implementation Details

Using the TensorFlow framework, we built the network and conducted experiments using NVIDIA GeForce RTX 2080 Ti. We used the Adam optimizer with a learning rate of 10−4, momentum coefficients β1 = 0.9 and β2 = 0.999, and the numerical stability constant ϵ = 10−8. The mini-batch size, α, and γ were set to 32, 60, and 10−4, respectively. The selected model maximized the detection accuracy on the validation set for 60 epochs.

Results on Frame-wise Detection

First, we conducted frame-wise detection with comparative CNN-based methods, which are specialized in exploring quantization artifacts on block-DCT or detecting fine-grained signals. Table 1(a) lists the performance metrics for each model for frame-wise detection, and we computed the ROC curve for a detailed performance analysis (see Figures 6(a) and 6(b)). Without the help of auxiliary features, the ACC, TNR, PRE, REC, F1-S, and AUC of the DHNet were 88.51%, 87.78%, 87.95%, 89.23%, 88.59%, and 0.9553, respectively. Importantly, DCT histogram-based approaches, including BarniNet, ParkNet, and our DHNet, effectively detected double compression artifacts on the decoded I-frame, achieving an ACC of over 85%. Particularly, among them, DHNet exhibited the best performance for all metrics, suggesting that the proposed method of adopting multiple DCT histograms better explores the statistical properties remaining in the DCT domain than comparative CNNs.
Next, to test the leverage of auxiliary features, we observed the performance change in the presence or absence of $F_a$. Providing the auxiliary features with the base matrix $Q_m$ and the quantizer scale $q_s$ to the dense layers enhanced the detection performance of DHNet and ParkNet (see Figures 6(c) and 6(d)). Importantly, the comprehensively analyzed results of Table 1(a) reveal that the DHNet-Q1 performs better than competitive methods. In addition, J-ALM-Q1 specialized in capturing the compression artifacts showed second-best performance. As the detection performance of CNN-Q1 outperforms that of CNN-Q2 overall, it is interpreted that $q_s$ is more helpful in identifying double compression than the elements constituting $Q_{m_2}$. Furthermore, accuracy tendency curves for training and validation in Figures 6(e) and 6(f) indicate that the adopting auxiliary features helps the model converge efficiently. These curves confirmed that the devised additional features facilitate single- and double-compression assessment.

**Results on GOP-wise Detection**

As mentioned in the subsection on datasets, videos for the main experiment were generated with a fixed GOP configuration; hence, peculiar quantization traces were repeatedly found in the decoded I-frames of the double-compressed video (Figure 3). To verify this experimentally using DHNet-Q1, we performed temporal detection on
each frame constituting a sample video (from the first to the 120\textsuperscript{th} frame). As displayed in Figures 6(i) to 6(q), on both single- and double-compressed videos, the prediction value peaks were generated periodically in GOP size intervals. In particular, the detection performance relatively deteriorated in inference to the decoded P-frames.

Next, we conducted GOP-wise detection on the test videos generated with the aligned GOP structure. Noting the temporal detection results according to the frame index, only decoded I-frames extracted from $\Phi$ GOPs on a given video were provided to the pretrained model. In this experiment, the top seven models for frame-wise detection were employed, and the majority voting mechanism was applied to the $\Phi$ prediction results. When $\Phi$ was set to 5 and 10, the detection ACC values of the DHNet-Q1 were 95.52% and 97.14%, respectively, as presented in Table 1(b). As the number of I-frames provided to each model increased (i.e., $\Phi = 1 \rightarrow \Phi = 5, 10$), the number of correct predictions for the corresponding true class increased.

Furthermore, we performed a performance comparison with the non-CNN approach, MF-LDA\textsuperscript{6} (see Figure 7). Unlike the CNN-based approaches to testing with only a single trained model, MF-LDA requires classifiers for each parameter and a combination of a single compression and double compression pair (requiring 12 classifiers for MF-LDA). The Markov features used by MF-LDA exhibited degraded performance in discriminating the statistical differences between specific pairs, and our work demonstrated outstanding performance overall.
To verify the effectiveness of multiple DCT histograms, we performed an ablation study on the combination of the histograms of $F_{h_4}$, $F_{h_8}$, and $F_{h_{16}}$. Seven combinations of DCT histograms were used, and the stream composition of the DHNet was adjusted according to the number of input features. When three types of DCT histograms were input into the DHNet, it displayed the best accuracy of 88.51%. For other combinations, accuracy values of less than 88% were observed. As the number of histograms provided to the DHNet decreased, the detection performance tended to deteriorate.

As illustrated in Figure 3, a non-aligned GOP structure may occur on double compression in videos. We experimented to determine the double compression of a given video using only the first decoded I-frame to cover this issue. The pretrained DHNet-Q1 without retraining was exploited, and videos were generated for the following GOP configurations $(g_1, g_2)$: $(5, 6)$, $(6, 5)$, $(6, 7)$, and $(7, 6)$. The average accuracy value was measured at 92.45%, which is 0.08% lower performance than the results for the aligned GOP structure (see Table 1(a)). The traces of the double quantization considered in the model training phase are left in the first I-frame of the video; thus, it is possible to handle the assumed case of a non-aligned structure.

In this article, we proposed a novel DHNet that identifies double MPEG-4 compression. For MPEG-4, block-DCT-based double compression with a different quantization scale inevitably causes rounding errors, leaving distinguishable footprints against single compression. Therefore, we devised multiple DCT histograms suitable for capturing peculiar artifacts in the DCT domain. Furthermore, performance improvement was achieved by providing auxiliary features to dense layers. The results revealed that this work is superior to the competitive methods, including CNN and non-CNN methods, in terms of various objective metrics. In the future, we will investigate the efficacy of the devised discriminant features against various compression standards that utilize block-based quantization.
dct coefficients,” *IET Inf. Secur.*, vol. 13, no. 1, pp. 1–6, 2019.

13. X. Jiang, Q. Xu, T. Sun, B. Li, and P. He, “Detection of hevc double compression with the same coding parameters based on analysis of intra coding quality degradation process,” *IEEE Trans. Info. Forens. Security*, vol. 15, pp. 250–263, 2019.

14. Q. Xu, X. Jiang, T. Sun, and A. C. Kot, “Detection of hevc double compression with non-aligned gop structures via inter-frame quality degradation analysis,” *Neurocomputing*, vol. 452, pp. 99–113, 2021.

15. S.-H. Nam et al., “Deep convolutional neural network for identifying seam-carving forgery,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3308–3326, 2021.

16. M. Yoon et al., “Frame-rate up-conversion detection based on convolutional neural network for learning spatiotemporal features,” arXiv:2103.13674, 2021.

17. M.-J. Kwon, I.-J. Yu, S.-H. Nam, and H.-K. Lee, “Cat-net: Compression artifact tracing network for detection and localization of image splicing,” in *proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 375–384.

18. J. Y. Lin et al., “Mcl-v: A streaming video quality assessment database,” *J. Vis. Commun. Image Represent.*, vol. 30, pp. 1–9, 2015.

19. C. Montgomery et al., “Xiph. org video test media (derf’s collection), the xiph open source community, 1994,” Online, https://media.xiph.org/video/derf.

Seung-Hun Nam is currently working at NAVER WEBTOON AI, South Korea. He received the Ph.D. degree from the School of Computing, KAIST, South Korea, in 2020. His research interests include digital watermarking and multimedia forensics. Contact him at shnam1520@gmail.com.

Wonhyuk Ahn is currently working at NAVER WEBTOON AI, South Korea. He received the Ph.D. degree from the School of Computing, KAIST, South Korea, in 2021. His research interests include deep learning and multimedia content protection. Contact him at whahnize@gmail.com.

Myung-Joon Kwon is currently pursuing the Ph.D. degree from the School of Electrical Engineering, KAIST, South Korea. He received the M.S. degree from the School of Computing, KAIST, South Korea, in 2021. His research interests include adversarial attacks, computer vision, and machine learning. Contact him at mjkwon2021@gmail.com.

Jihyeon Kang is currently working at NAVER WEBTOON AI, South Korea. He received the Ph.D. degree from the Graduate School of Information Security, KAIST, South Korea, in 2021. His research interests include machine learning and computer vision. Contact him at mangorism@gmail.com.

In-Jae Yu is currently working at Samsung Electronics, South Korea. He received the Ph.D. degree from the School of Computing, KAIST, South Korea, in 2021. His research interests include machine learning and computer vision. Contact him at myhome98304@gmail.com.