End-to-end double JPEG detection with a 3D convolutional network in the DCT domain


Detection of double JPEG compression is essential in the field of digital image forensics. Although double JPEG compression detection methods have greatly improved with the development of convolutional neural networks (CNNs), they rely on handcrafted features such as discrete cosine transform (DCT) histograms. In this Letter, the authors propose an end-to-end trainable 3D CNN in the DCT domain for double JPEG compression detection. Moreover, they also propose a new type of module, called feature rescaling, to insert the quantisation table into the network suitably. The experiments show that the proposed method outperforms state-of-the-art methods.

Introduction: Blind image forensics aims to verify the authenticity or integrity of a digital image, without any signatures or watermarks. The mainstream approach of blind image forensics is to assume a specific type of image manipulation, such as copy-move, splicing, and retouching, and to design a detecting algorithm to target it. Although the algorithms perform well for targeting image manipulations, all methods regarding different manipulation types should be used to authenticate a single suspicious image, which leads to false alarms in multiple testing. The problem further complicated if the forger mixes multiple manipulations not only to disguise the tampered region visually, but also to avoid detection by the forgery detection methods. The double JPEG detection method refers to the technique of classifying a given image between a single compressed JPEG image and a double compressed JPEG image. Rather than focusing on a specific manipulation, it regards traces of double JPEG as abnormal image processing because JPEG compression is usually done once when taking the picture itself, and double JPEG compression happens when resaving the acquired image. Since JPEG encoding is widely used and double JPEG detection methods can be used to investigate suspicious image regardless of the type of manipulation, significant attention has been devoted to the development of double JPEG detectors.

Double JPEG compression leaves peculiar artefacts in a discrete cosine transform (DCT) domain; thus most existing double JPEG detection methods rely on the statistic of DCT coefficients, such as distribution of first significant digits and DCT histograms. Although adopting convolutional neural networks (CNNs), existing approaches employ the histogram as an input. Wang and Zhang [1] first proposed CNN-based double JPEG detection using a 1D DCT histogram feature of ten AC components. In [2], Barni et al. improved upon the method in [1] with a 2D DCT histogram of all DC and AC components, and it outperformed all previous methods based on handcrafted features. Park et al. [3] presented a more practical and tough scenario where images were compressed by mixing a 1120-quantisation table randomly, which were found in images requested from the forensic website. The authors optimised parameters for histogram generation given in [2] and inserted quantisation tables into the network for double JPEG compression of mixed quality factors. [3] achieved the current state-of-the-art performance for double JPEG detection. Although CNNs based on DCT histograms showed effective performance in double JPEG detection [2, 3], they lost information left in raw DCT coefficients when preprocessing the histogram. This leads to suppression of the network in learning fine-grain features. In this Letter, we propose end-to-end neural networks for double JPEG detection without histogram generation. By taking raw DCT coefficients, the proposed method finds the feature domain by itself and shows better performance than the state-of-the-art methods [2, 3].

Analysis on double JPEG: In JPEG compression, the image is divided into 8 × 8 pixel blocks. For each block B, the DCT is applied using the following formula:

\[ D_{ij} = \sum_{k,l=0}^{7} a_{ij}(k,l) B_{kl}, \]

where \( a_{ij}(k,l) = \frac{1}{\sqrt{2}} \sin(\pi k/8) \sin(\pi l/8) \cos(\pi (2l+1)/8) \) and \( w(k) = \frac{1}{\sqrt{2}} \) for \( k = 1 \) and \( w(k) = 1 \) otherwise. Quantised coefficients \( D^{\nu} \) are obtained by quantising \( D \) using a quantisation matrix \( Q \)

\[ D_{ij}^{\nu} = \text{round}(D_{ij}/Q_{ij}), \quad i, j \in [0, \ldots, 7]. \]

JPEG compression is done by storing \( D^{\nu} \) and \( Q \) in JPEG file using zig-zag ordering and Huffman coding. When decompressing the JPEG image, the dequantised DCT coefficients \( D^{\nu} \) are restored by \( D_{ij}^{\nu} = D_{ij}^{\nu} \times Q_{ij}. \) Then, the decompressed pixel block, \( B^{\nu} \), is obtained by inverse DCT (IDCT) with rounding and truncation to fit the range \([-128, 127]\)

\[ B^{\nu} = \text{truncate}((\text{IDCT}(D^{\nu}))). \]

Let us denote \( Q_{1} \) and \( Q_{2} \) by the quantisation tables used in first and second JPEG compression, respectively, and assume \( B^{\nu} \) is obtained using \( Q_{1}. \) When compressing \( B^{\nu} \) again with \( Q_{2} \), (1) and (2) are applied to \( B^{\nu}. \) Due to truncation and rounding in (3), the obtained \( D_{ij} \) in (1) usually lose their integer values, and are therefore not multiples of \( Q_{ij}^{1} \) but spread around these multiples. The coefficients concentrating on multiples of \( Q_{ij}^{1} \) are then quantised using \( Q_{ij}^{2} \) in (2), which creates a pattern in the histogram of values \( D_{ij}. \)

Therefore, most existing methods [1–3] rely on the histogram and its related features, even after adopting a CNN framework. Although the CNNs based on DCT histograms performed effectively for double JPEG detection [2, 3], they lose information left in raw DCT coefficients, when preprocessing the histogram, which leads to suppression of the network in learning fine-grain features.

Proposed method: An overview of the proposed method is shown in Fig. 1. In our scheme, we present a double JPEG compression detection method using 3D CNN learning in an end-to-end way. We also propose a new module, called feature rescaling, to effectively incorporate quantisation tables into the network; this rescales feature maps dynamically according to quantisation step \( Q_{ij}. \) We now describe each step in detail.

End-to-end double JPEG detection with a 3D convolutional network in the DCT domain

**Step 1: frequency splitting:** The proposed method uses raw dequantised DCT coefficients, \( D^{\nu} \). Given a JPEG image, we obtain \( D^{\nu} \) using \( D^{\nu} \) and \( Q \) stored in the JPEG file. JPEG images with a size of \( 256 \times 256 \) is a unit of decision for our detector, \( D^{\nu} \), a group of \( 8 \times 8 \) blocks as presented in Fig. 1, each consisting of the energy of 64 different frequencies. We divide the coefficients into the same frequency and get \( L_{ij} \) as follows:

\[ L_{ij}(k, l) = \text{round}(D^{\nu} + 8k + 8l), \quad i, j \in [0, \ldots, 7], \quad k, l \in [0, \ldots, 31]. \]

The elements of \( L_{ij} \) are the coefficients of the same frequency for each block and multiples of \( Q_{ij}. \) Moreover, if the image is doubly compressed, the traces induced by quantisation steps \( Q_{ij}^{1} \) and \( Q_{ij}^{2} \) are aggregated into \( L_{ij}. \) We reshape \( L \) of size \( 8 \times 8 \times 32 \times 32 \) into
Step 2: residual 3D convolutional neural networks: 3D CNN is devised for learning spatiotemporal features in video applications [4]. 3D CNN consists of 3D kernels; that learn temporal features as well as spatial information. We adopt the 3D CNN framework into our network as it offers the following advantages:

- Arranged coefficients \( L \) have inter-frequency (corresponding to temporal information in the video) features as well as spatial features.
- We can incorporate a quantisation table into the network more adequately using the proposed feature rescaling.

For simplicity, we refer to feature maps with a size of \( c \times l \times h \times w \), where \( c \) is the number of channels, \( l \) is length in number of frequencies, and \( h \) and \( w \) are the height and width of the feature, respectively. The input \( L \) defined in the previous section is a tensor of one channel, and 64 frequencies, and height and width are 32. We also refer to 3D kernel size by \( d \times k \times k \), where \( d \) is kernel inter-frequency depth, and \( k \) is the kernel spatial size. We now elaborate on the architecture shown in Fig. 1. Considering that traces of double JPEG compression typically remains in intra-frequency statistics and most existing methods rely on it, we make the network focus the features on intra-frequency at first. Specifically, we construct the first layer of the network by using 30 kernels of size \( 1 \times 5 \times 5 \) with stride \((1, 1, 1)\), which extracts only the intra-frequency relation information.

The output is then followed by three blocks. We define the block as in Fig. 1, to learn inter-frequency features together with intra-frequency features using 3D convolutions. Each block employs two types of units: one expansion unit and three residual units. The expansion unit uses one kernel of \( 1 \times 1 \times 1 \) with specified number of filters and stride denoted in block. It is used to scale up the number of channel \( c \) before the residual unit. The residual unit consists of two kernels of \( 3 \times 3 \times 3 \) with stride \((1, 1, 1)\), and we inject a skip connection between the beginning and end of the unit, which is summed with the output of the second convolution. All feature maps are pre-activated by batch normalisation and rectified linear unit before the convolution.

Inspired by Broumand et al. [2] showing effective performance in steganalysis by unpooling the feature maps in early layers to capture subtle signals modified by steganography, we do not reduce the spatial size of the feature until the end of the block 1 with stride 1 for all directions. From block 2, we reduce the spatial size of feature maps by adopting stride \((1, 2, 2)\) at the expansion unit.

Step 3: feature rescaling: The method proposed in [3] concatenated a vectorised quantisation table into all fully connected (FC) layers, while learning inter-frequency features. In the end, the output feature of block 3 \( f_{3o} \) had a shape of \( 90 \times 64 \times 8 \times 8 \). We would like to note that the \( f_{3o} \) is extracted from dequantised coefficients, so \( f_{3o} \) is formed in the level of dequantised coefficients. To let the network uses the signal in the level of quantised coefficient, we divide the \( f_{3o} \) with vectorised quantisation table \( Q \) and get \( f_{3o}/Q \), by imitating the quantisation operation in JPEG compression as follows:

\[
f_{3o}/Q(i, j, k, l) = f_{3o}(i, j, k, l) / Q_{ij},
\]

where \( i, j, k, l \) are indexes of channel, frequency, height, and width, respectively. We concatenate \( f_{3o} \) and \( f_{3o}/Q \) along the channel dimension to get \( f_{3o} = [f_{3o}, f_{3o}/Q] \). Finally, \( f_{out} \) is pooled using global average pooling (GAP) and connected with the FC layer with one neuron for classification.

Experiments: Park et al. [3] analysed the characteristics of suspicious images requested from the forensic website. They found that 41.77% of images used a non-standard quantisation table and 1170 quantisation tables were found. Based on this fact, they generated 1,140,430 images with random quantisation tables using public raw image datasets to make the scenario more practical. We experimented with the proposed method and related works using this dataset with 1,026,387 images to train 114,043 images for testing. We evaluate our method and compare it with [2, 3]. The models were implemented with PyTorch 1.1.0 trained on the dataset described above, using an Adam optimiser with an initial learning rate of 0.001.

We reported error rates on the test sets for all models in Table 1 with the number of parameters. We also present the result of the proposed method without feature rescaling, where \( f_{3o} \) is directly pooled with GAP to show the effectiveness of the architecture itself. For [2, 3], we construct the histogram using raw DCT coefficients instead of calculating from the pixels domain, which is the original in [2, 3], for a fair comparison. Barni et al. achieved an error rate of 13.17%, which is 3.36% lower than extracting a DCT histogram from the pixel domain. This indicated that network is interrupted due to rounding and truncation errors arose from decompensation. However, Park et al. showed a 7.4% error rate, which is similar to the performance reported in [3], and did not benefit from using a direct DCT coefficient. It suggests that the quantisation table information is quite a strong feature.

<table>
<thead>
<tr>
<th>Method</th>
<th>No. of Parameters</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barni et al. [2] (from pixels)</td>
<td>0.59 M</td>
<td>16.53</td>
</tr>
<tr>
<td>Barni et al. [2]</td>
<td>0.59 M</td>
<td>13.17</td>
</tr>
<tr>
<td>Park et al. with q. matrix [3]</td>
<td>16.8 M</td>
<td>7.4</td>
</tr>
<tr>
<td>proposed w/o feature rescaling</td>
<td>1.66 M</td>
<td>8.84</td>
</tr>
<tr>
<td>proposed with q. matrix</td>
<td>1.66 M</td>
<td>8.42</td>
</tr>
<tr>
<td>proposed with feature rescaling</td>
<td>1.66 M</td>
<td>6.14</td>
</tr>
</tbody>
</table>

The proposed method without feature rescaling classified single or double JPEG with an 8.84% error rate, which is lower than Barni et al.’s by 4.33%. We want to note the performance is close to that of Park et al. by 1.44%, with fewer parameters and without quantisation table information. This suggests that the proposed method of adopting 3D convolution with end-to-end learning can capture features that a DCT histogram cannot.

Fig. 2 Localisation results on various manipulations: copy-move, content-aware interpolation, splicing, and splicing from top to bottom

- **a**: Normal
- **b**: Manipulated
- **c**: Ground truth
- **d**: Localisation result of proposed method

Next, we consider options for incorporating the quantisation table into the network. Park et al. concatenated vectorised quantisation tables into all three FC layers which improved the error rate. However, this strategy causes the network to heavily rely on the FC layer, generating more than 10M in parameters. Instead, we adopt only one FC layer to decrease dependence on the FC layer. We evaluated the proposed network with a vectorised quantised table in the FC layer as conducted by Park et al. [3] and compared it with it using feature rescaling. The results showed that a vectorised quantised table in FC layer resulted...
in minor progress (0.41%), but feature rescaling allowed the network to achieve an error rate of 6.14% with a 2.7% margin, achieving the state-of-the-art performance.

Fig. 2 shows the localisation results on various manipulations using the proposed method. We did the qualitative experiment with the following procedures. The RAW images are first compressed with a random quantisation table among 1120 tables to get a single original JPEG as in Fig. 2a. This usually happens when taking the picture as post-processing to reduce the camera storage. Then, the images are manipulated by copying an object and pasting it into a different location, erasing the banner using content-aware interpolation, or slicing an object from another image and pasting into it. Lastly, the manipulated images are saved with another different random quantisation table (Fig. 2b). Fig. 2c represents the ground truth of the manipulated region.

We can use the proposed double JPEG detection method to localise the manipulated regions, because un-manipulated regions exhibit the characteristics of double JPEG and manipulated regions lose the characteristics. The suspicious image is split into 256 × 256 patches with a stride of 16 × 16, and the proposed method predicts whether a given patch is doubly or singly compressed. In this way, we can acquire the prediction map as in Fig. 2d, where the red and blue colours represent single and double JPEG patches, respectively. The proposed method localised the manipulated regions successfully regardless of the type of manipulation.

**Conclusion**: In this Letter, we proposed an end-to-end 3D CNN in a DCT domain to detect double JPEG compression of mixed quality. The proposed method consists of frequency splitting to arrange the coefficients into the same frequency, residual 3D CNN to capture intra- and inter-frequency features, and feature rescaling to incorporate quantisation table information adaptively. The experimental results showed that the proposed method outperformed the state-of-the-art methods.

**Acknowledgment**: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2017-0-01671, Development of high reliability image and video authentication service for smart media environment).

© The Institution of Engineering and Technology 2020
Submitted: 9 August 2019 E-first: 30 October 2019
doi: 10.1049/el.2019.2719
One or more of the Figures in this Letter are available in colour online.
W. Ahn, S.-H. Nam, M. Son, H.-K. Lee and S. Choi (School of Computing, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea)
✉ E-mail: whahnize@kaist.ac.kr

**References**