Self-Supervised Image Local Forgery Detection by JPEG Compression Trace

Xiuli Bi, Wuqing Yan, Bo Liu, Bin Xiao*, Weisheng Li, Xinbo Gao

Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
{bixl, boliu, xiaobin, liws, gaoxb}@cqupt.edu.cn, s210201122@stu.cqupt.edu.cn

Abstract

For image local forgery detection, the existing methods require a large amount of labeled data for training, and most of them cannot detect multiple types of forgery simultaneously. In this paper, we firstly analyzed the JPEG compression traces which are mainly caused by different JPEG compression chains, and designed a trace extractor to learn such traces. Then, we utilized the trace extractor as the backbone and trained self-supervised to strengthen the discrimination ability of learned traces. With its benefits, regions with different JPEG compression chains can easily be distinguished within a forged image. Furthermore, our method does not rely on a large amount of training data, and even does not require any forged images for training. Experiments show that the proposed method can detect image local forgery on different datasets without re-training, and keep stable performance over various types of image local forgery.

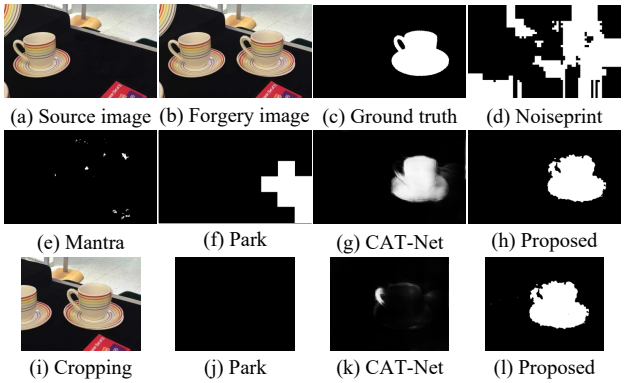


Figure 1: An example of local forgery and corresponding localization results of four types of detection methods.

Introduction

In recent years, researchers have proposed many methods to verify the integrity and authenticity of digital images. Image forgery detection is to judge whether an image has been forged or not without any prior knowledge, which can be divided into two categories according to the forgery scope: global forgery detection and local forgery detection. The former mainly detects whether an image is manipulated by some image operations such as contrast enhancement, image filtering, and image compression or not. The latter is to detect whether the content of an image is changed. The local forgery mainly includes copy-move, splicing, and object removal. In this paper, we focused on local forgery detection.

Splicing is the most common local forgery, which usually cuts out objects from one or more other images and then pastes them onto the target image. For the splicing detection and location, because the tampered and un-tampered regions are from different sources, that can be detected by finding the distinction between them (Bappy et al. 2017; Bondi et al. 2017; Bunk et al. 2017; Liu and Pun 2018; Bi et al. 2019; Mayer and Stamm 2019). A copy-move forgery denotes an image where part of its content has been copied and pasted within the same image. Hence, the characteristics of the tampered regions are similar to the non-tampered regions. Various existing methods are explored to extract the similarity from image blocks, key points or the network-learned features (Li and Zhou 2018; Wu, Abd-Almageed, and Natarajan 2018; Zhong and Pun 2019; Zhu et al. 2020; Bilal et al. 2021). For object removal, after removing the target, the left blank region will be filled by an inpainting algorithm. Based on the research on the inpainting algorithm, many effective detection methods have been proposed (Liang et al. 2015; Li, Luo, and Huang 2017; Zhang et al. 2018; Li and Huang 2019; Wang, Niu, and Wang 2021). Excluding the detection methods for a specific local forgery, there are also well-designed algorithms that can detect multiple local forgeries, such as Mantra (Wu, AbdAlmageed, and Natarajan 2019), Noiseprint (Cozzolino and Verdoliva 2019), and so on (Zhou et al. 2018; Bappy et al. 2019; Rao and Ni 2021).

According to the summarization of requested images through the forensic website over two years (Park et al. 2018), the JPEG format was found to be the most requested (77.95%), followed by PNG (20.67%). Therefore, more and more researchers focus on the forensics of JPEG images. For local forgery detection in JPEG images, most methods mentioned above perform poorly or even fail, as shown in Figure 1. (d-e). The reason is that JPEG compression is a lossy compression, and the tampering traces will disappear

*Corresponding author

along with the image content during compressions. Therefore, researchers need to propose the local forgery detection method based on analyzing the compression process of JPEG images. These proposed detection methods for JPEG images can be divided into the traditional detection methods (Fu, Shi, and Su 2007; Lin et al. 2009; Bianchi and Piva 2012; Wang, Dong, and Tan 2014; Yang et al. 2020; Niu et al. 2021) and the CNN-based detection methods.

Among these CNN-based detection methods, (Wang and Zhang 2016) distinguishes the singly compressed regions and doubly compressed regions by classifying DCT histograms; (Amerini et al. 2017) improves the performance of frequency domain-based CNN; Park (Park et al. 2018) collected real-world JPEG images from the image forensic service and obtained 1120 quantization tables to generate a JPEG dataset to train the network; (Deng et al. 2019) improves the detection performance in difficult situations by designing modules to automatically extract multiple features from DCT histograms of JPEG images. CAT-Net (Kwon et al. 2021) combines the spatial and DCT frequency stream to learn forensic features of compression traces.

Although these CNN-based detection methods have made great contributions to the forensics of JPEG images, their methods need large training sets. Meanwhile, since they rely on the 8×8 DCT coefficient matrices, they cannot perform the pixel-level detection or work under a cropping attack, as shown in Figure 1. Furthermore, their method works for specific situations, such as the combined single and double compressions. Based on these issues, this paper proposed a self-supervised image local forgery detection method by JPEG compression trace. The main contributions of our proposed method are listed as follows:

- The proposed JPEG compression trace extractor can directly extract JPEG compression traces from the whole image, rather than dividing images into 8×8 blocks to analyze DCT coefficients.
- We proved the detection of different types of local forgery can convert to distinguish compression chains of tampered and un-tampered regions, so they can be detected in the same way.
- The proposed self-supervised detection method does not rely on a large amount of training data, and even does not require any forged images for training.

Extensive experiments show that the proposed method has a good ability to detect various local forgeries in JPEG images and can resist cropping attacks well.

JPEG Compression Trace Extractor

JPEG Compression Trace

In the JPEG compression pipeline, each 8×8 image block $f(i,j)$ of an image I will be transformed by

$$F(u,v) = DCT(f(i,j)). \tag{1}$$

Where, $DCT()$ is two-dimensional Discrete Cosine Transform (DCT), and $F(u,v)$ is the DCT coefficient matrix. Then, a 8×8 quantization table Q will quantize the DCT coefficient matrix $F(u,v)$ by

$$F_Q(u,v) = \lfloor \frac{F(u,v)}{Q(u,v)} \rfloor. \tag{2}$$

Where, $\lfloor \cdot \rfloor$ means rounding down. The quantization table Q corresponds to a quality factor QF. The calculation of the quantization table $Q(u,v)$ defined by JPEG standard is

$$Q(u,v) = \left[\left(Q(u,v)^T \times \alpha\left(QF\right) + 50 \right) / 100 \right]. \tag{3}$$

Where $Q(u,v)^T$ is the standard JPEG quantization matrix given by Joint Photographic Experts Group (JPEG). Once QF is chosen, $\alpha\left(QF\right)$ will be calculated by

$$\alpha\left(QF\right) = \begin{cases} 5000 \,/\, QF, & 1 \leqslant QF < 50 \\ 200 - 2QF, & 50 \leqslant QF < 100 \end{cases} \tag{4}$$

If the rounded-off part is regarded as the truncation error $e(u,v) \in (0, Q(u,v))$ caused by quantization procedure, Eq. (2) can be rewritten as

$$F_Q(u,v) = F(u,v) - e(u,v). \tag{5}$$

When $F_Q(u,v)$ is decompressed, the inverse $iDCT()$ transformation will be applied on $F_Q(u,v)$ to obtain $f'(i,j)$ by

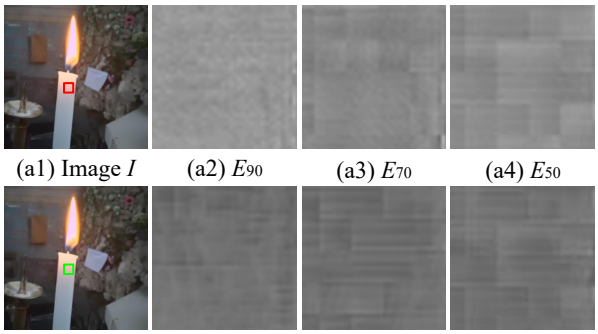
$$f'(i,j) = iDCT(F_Q(u,v)) = iDCT(F(u,v) - e(u,v)). \tag{6}$$

Since $iDCT()$ transformation satisfies the linear invariant property, the compressed block $f'(i,j) = f(i,j) + iDCT(-e)$, and we can denote the compressed image as

$$I_{QF} = I + E_{QF}, \tag{7}$$

where, E_{QF} represents $iDCT(-e)$ of all the 8×8 blocks.

It can be seen in Eq. (5) and Eq. (6), the error $e(u,v)$ exists in the whole process of compression and decompression. As shown in Figure 2, we compressed an image I by using different QF and subtracted the compressed images I_{QF} in the green boxes from the uncompressed image in the red box. The subtracted results are the error E_{QF} and are shown in Figure 2. (a2-a4). We can see that the error E_{QF} exhibits block artifacts and is affected by the image content. When the QF value is smaller, the error E_{QF} is greater, and the block artifacts are more obvious.



(a1) Image I (a2) E_{90} (a3) E_{70} (a4) E_{50}

(b1) Image I_{QF} (b2) J_{90} (b3) J_{70} (b4) J_{50}

Figure 2: The true and learned JPEG compression traces. (a2-a4) are the truncation errors $E_{QF=90,70,50}$ generated by subtracting the red box in the image I from the green box in I_{QF}. (b2-b4) are the learned JPEG compression trace $J_{QF=90,70,50}$ from the corresponding JPEG images.

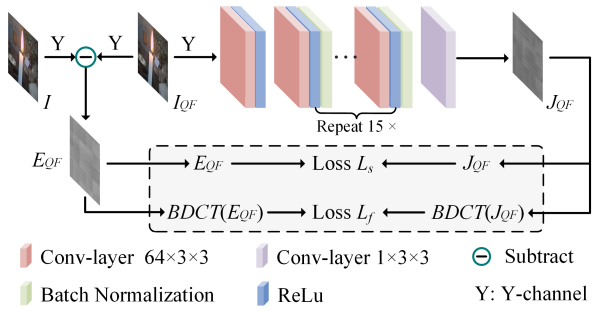


Figure 3: The JPEG compression trace extractor.

JPEG Compression Trace Extractor

Although the block artifacts caused by E_{QF} have been explored in the forensic methods (Farid 2009; Bianchi and Piva 2011, 2012; Zhang et al. 2020; Amerini et al. 2017; Kwon et al. 2021; Rao and Ni 2021), they use statistics or learn from the histogram distribution of 8×8 blocks' DCT coefficients, which makes them not robust to cropping attacks and can not achieve pixel-level localization. Since E_{QF} is the fundamental cause of the JPEG compression trace, we hope to remove the influence of image content through deep learning and to learn the distribution of the truncation error E_{QF} from the images directly. Compared to the previous methods, our method is a new perspective on JPEG image forgery detection, which fundamentally solves the problem of cropping attacks and improves the localization performance.

For extracting E_{QF} in Eq. (7), we were inspired by the residual learning proposed in (Zhang et al. 2017) and designed a JPEG compression trace extractor, as shown in Figure 3. For making the JPEG compression trace extractor learn the error E_{QF} directly from the input compressed image I_{QF}, we designed its loss function L as

$$L = (1-\lambda)\, L_s + \lambda \cdot L_f. \tag{8}$$

Where, L_s is spatial domain loss, L_f is frequency domain loss. Since the RGB image is converted to YCbCr color space and then compressed in separate channels in JPEG compression, and Y luminance channel contains the most information of the image, we selected the Y channel for learning the compression error E_{QF} through the network.

In the spatial domain, L_s is defined as

$$L_s = \frac{1}{N}\sum_{i=1}^{N} \|J_{QF}[i] - E_{QF}[i]\|^2, \tag{9}$$

where, N indicates the batch size of the input, $J_{QF}[i]$ represents the output by JPEG compression trace extractor. For the frequency domain, we can directly conclude E_{QF} by Eq. (5) and Eq. (6). We perform 8×8 block DCT transformation (BDCT) on the error E_{QF} to obtain the true truncation error $BDCT(E_{QF})$. Then, the loss L_f between the network output and $BDCT(E_{QF})$ in the frequency domain can be calculated by

$$L_f = \frac{1}{N}\|BDCT(J_{QF}[i]) - BDCT(E_{QF}[i])\|^2. \tag{10}$$

The extractor continuously narrows the gap between its output and the error E_{QF} by inputting the original image I and the compressed image I_{QF} in pairs, finally, it can extract JPEG compression trace directly from images. In Figure 2. (b2-b4), we can see that the extracted JPEG compression trace J_{QF} owns the same block artifact of E_{QF}.

Self-Supervised Local Forgery Detection by JPEG Compression Traces

Local Forgery under JPEG Image

For analyzing image local forgery detection in JPEG images, we firstly propose the concept of a compression chain that describes the compression history of a JPEG image.

$$I_{QF_1\_QF_2\_\ldots\_QF_n} = QF_n \bullet (\ldots QF_2 \bullet (QF_1 \bullet (I))), \tag{11}$$

When we consider local forgery in JPEG images, it is a JPEG image that is tampered with and saved in JPEG again. We assume the source image's compression chain is $QF_1\_QF_2\_\ldots\_QF_n$. As shown in Figure 4, the dotted lines indicate the original trace of the tampered regions, and the solid red lines indicate the QF_{n+1} grid. Whether the tampered regions come from the other images, the same source image, or generated by an inpainting algorithm, the tampered regions are very likely divided into new 8×8 blocks and compressed again, which will cause a non-aligned overlapping compression. Therefore, we consider that the tampered regions in local forgery own a new compression chain that restarts from QF_{n+1}. However, the un-tampered regions underwent an overlapping compression as shown by the overlapping solid blue and red lines, which makes the compression chain continuous: $QF_1\_QF_2\_\ldots\_QF_n\_QF_{n+1}$. We analyzed the results extracted by the JPEG compression trace extractor. The traces of the tampered and un-tampered regions are inconsistent, as shown in Figure 4. Based on this insight, since all local forgery will make the tampered and the un-tampered regions own different compression chains, and different compression chains will cause inconsistency of compression traces, this phenomenon makes it possible for us to detect different types of local forgery in the same way.

Self-Supervised Local Forgery Detection

Although we know the compression traces of the tampered and the un-tampered regions are inconsistent, it is difficult to locate them directly by the output of the JPEG compression trace extractor, as shown in Figure 4. Inspired by the contrastive learning SimCLR (Chen et al. 2020), we propose a self-supervised image local forgery detection network shown in Figure 5. We regard distinguishing the extracted JPEG compression traces of different compression chains as a classification task, transferring the extracted compression traces to other spaces to strengthen their difference. In this way, the proposed method can detect various local forgeries with different JPEG compression cases.

We have known that the local forgery detection in JPEG images is to distinguish between different compression chains. However, we cannot accurately distinguish each

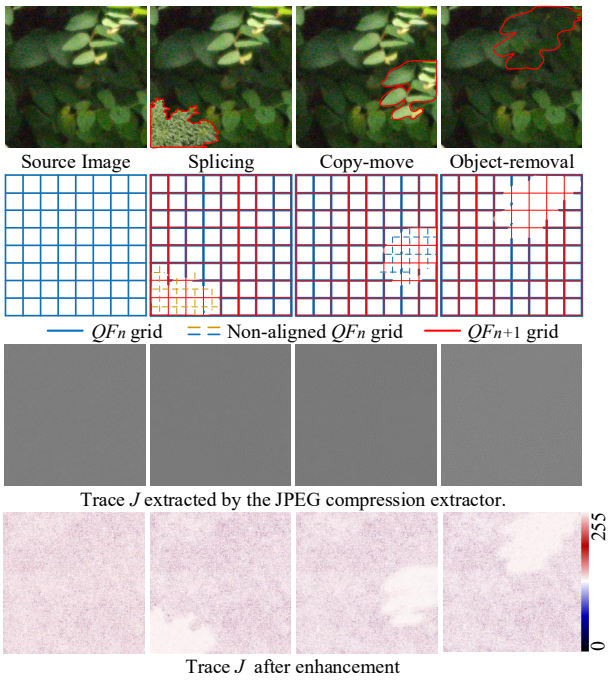


Figure 4: The analysis of three types of image local forgery under JPEG format. The red box on the forgery image is the ground truth of the tampered regions. We normalized the value of J to [0, 1] and selected the appropriate color from LUT (Look-Up-Table), then manually adjusted the brightness and contrast to show the tampered regions obviously.

chain because there are tens of thousands of combinations. Therefore, we first theoretically analyze the error caused by multiple compressions. Based on the definition of E_{QF} caused by a single compression with QF in subsection 2.1, we define the error of double compressions as

$$E^2 = iDCT\left(F(u,v) - \lfloor F_{Q_1}(u,v) \cdot \frac{Q_1(u,v)}{Q_2(u,v)} \rfloor\right). \quad (12)$$

E^2 is the error caused by $F(u,v)$ after $Q_1(u,v)$ quantization and then $Q_2(u,v)$ quantization. The process of JPEG compression is the same as the modulo operation; the truncation error E^2 is equivalent to the remainders in integer division, so we introduce the concept of the Residual System. For a sequence of $X = 1, 2, .., n \times m$, we took the modulus of n, m ($n < m$) to get two residual systems R_n and R_m, compared the corresponding elements of the two residual systems, we can conclude the probability that the element in the residual system R_n is greater than the element in R_m by

$$P(R_n > R_m) = \frac{n-g}{2m}. \quad (13)$$

Where, $g = gcd(m,n)$ is the greatest common divisor of m and n. As shown in Table 1, when $n = 3$ and $m = 5$, the bold numbers indicate $R_3 > R_5$. Among them, $15 \times [1 - P(R_3 > R_5)] = 12$ points $R_3 \leq R_5$, these 12 points are first modulo 3 and then modulo 5 (R_{3_5}), which is equivalent to discarding R_5, and they have the same effect as

X_i	1	2	3	4	**5**	6	7	8	9	**10**	**11**	12	13	14	15
R_3	1	2	0	1	**2**	0	1	2	0	**1**	**2**	0	1	2	0
R_5	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0
R_{3_5}	1	2	3	4	**5**	1	2	3	4	**5**	**6**	2	3	4	0
R_{5_3}	1	2	3	4	2	3	4	5	6	1	2	3	4	5	0

Table 1: The residual system of R_3, R_5, R_{3_5}, R_{5_3} in [1,15].

R_5. There are only $15 \times P(R_3 > R_5) = 3$ points where R_3 is bigger than R_5, and only these three points R_{3_5} are different from R_5. On the contrary, these points are first modulo 5 and then modulo 3 (R_{5_3}), so it will lose more numbers than R_5, but overall, R_{5_3} is closer to R_5 rather than R_3. By analyzing the truncation error caused by the quantization process of double compressions and combining the examples of the residual system, we can conclude that the impact of the compression chains on the image is related to the compression order but mainly depends on the minimum QF, which has the maximum quantization step. Therefore, the key to distinguishing compression chains is that the single quality factors can be well distinguished.

Suppose we want the network to distinguish N different $QF \in \{Q_1, Q_2, ...Q_k, ...Q_N\}$. As shown in Figure 5, we randomly selected N uncompressed images I and cropped them into $4N$ image patches. Then 4 patches are randomly chosen as a group P_k and then compressed by the same Q_k. There are N compression patch groups that correspond to N different QF. This procedure prevents the network from classifying image patches into one class based on the same source image rather than the same QF. The Y channels of compression patch groups P_k were fed to the JPEG compression trace extractor to extract the trace groups J_k. For each trace $J_{k[a]}$ ($a \in \{1,2,3,4\}$) in the group k, we intent to make it close to the same class J_{k+} and alienate other classes J_{k-}, which means $sim(J_{k[a]}, J_{k+}) \to 1$ and $sim(J_{k[a]}, J_{k-}) \to 0$. Therefore, the loss function is:

$$l(k, k[a]) = -\log \frac{sim(J_{k[a]}, J_{k+})}{sim(J_{k[a]}, J_{k+}) + sim(J_{k[a]}, J_{k-})}. \quad (14)$$

The similarity of different traces $J_{i[a]}$, $J_{j[b]}$ is computed by

$$sim(J_{i[a]}, J_{j[b]}) = softmax(d(J_{i[a]}, J_{j[b]})) = \frac{e^{-d(J_{i[a]}, J_{j[b]})}}{\sum_{i[a] \neq j[b]} e^{-d(J_{i[a]}, J_{j[b]})}}, i,j \in [1,N]; a,b \in [1,4]. \quad (15)$$

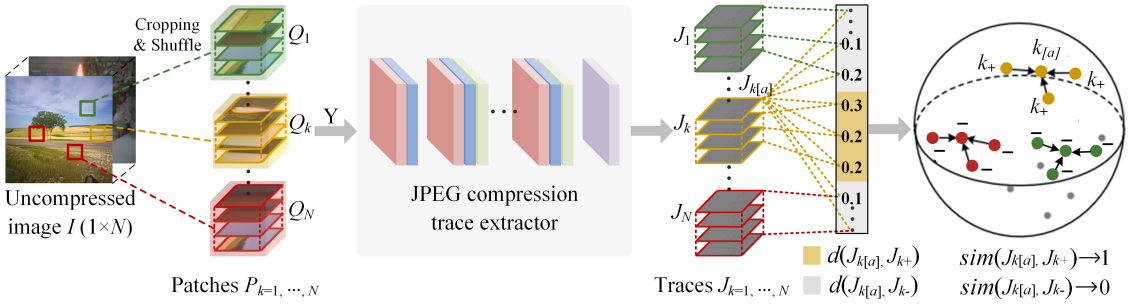
Where, the distance between $J_{i[a]}$, $J_{j[b]}$ is the squared Euclidean distance, and is calculated by

$$d(J_{i[a]}, J_{j[b]}) = \left\| J_{i[a]} - J_{j[b]} \right\|^2. \quad (16)$$

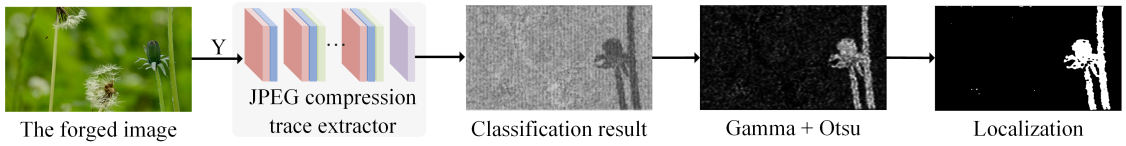
Finally, the total loss is

$$L = \sum_{k=1}^{N} \sum_{a=1}^{4} l(k, k[a]). \quad (17)$$

Through contrastive learning, $l(k, k[a])$ and L will decrease towards 0. The JPEG compression trace extractor will increase the discrimination between the compression traces



(a) The self-supervised training process.



(b) The process of forgery localization.

Figure 5: The framework of the proposed Self-Supervised Image Local Forgery Detection.

and can detect various local forgeries and different JPEG compression cases in which the tampered images are saved.

Experiments

Dataset and Metric

Experimental Dataset. Compared to the other local forgery detection methods as listed in Table 2, our method does not need annotated labels or large numbers of images for training. We only used 200 TIFF-formated images randomly selected from the ALASKA (Ruiz et al. 2021) dataset.

To verify whether our method can detect three classical types of local forgery well or not, we chose a variety of datasets that include splicing (SP), copy-move (CM), and object-removal (OR). The Spliced CoCo dataset is proposed in CAT-Net (Kwon et al. 2021), which only has splicing forgery images. Coverage (Wen et al. 2016) is a classic dataset for copy-move forgery detection. The Korus dataset (Korus and Huang 2016) is realistic images taken by four different cameras with high resolution and contains all of the three types of image local forgery.

Similar to most methods (Park et al. 2018), we produced the S-M situation, which means the tampered regions are with single compression, and the un-tampered regions are with multiple compressions. Firstly, we generated single JPEG images using different QFs. Second, we tampered with parts of these images. Third, we saved the images in JPEG format. Moreover, we also made the other two situations to simulate the actual conditions. If we save the tampered images as an uncompressed format in the third step, the tampered and the un-tampered regions are both with single compression (S-S); if we save it in JPEG format multiple times with different QFs, the tampered the un-tampered regions are with both multiple compressions (M-M).

Experimental Metric. The performance was evaluated by

$$F1 = \frac{2 \times Precison \times Recall}{Precison + Recall} \times 100\%, \qquad (18)$$

Train Set			Applicable Situation	
Method	Number	QF	Type	Situation
Noise.	2000	∼	SP, CM	∼
Mantra	102028	∼	SP, CM, OR	∼
Park	18946	1120	SP	S-M
CAT-Net	968683	153	SP	S-M
Proposed	200	50	SP, CM, OR	S-S, S-M, M-M

Table 2: The training sets and applicable situation of different methods. (∼ means unaffected by this option.)

where $Precision$ is the ratio of the correctly detected regions to all detected regions. $Recall$ is the ratio of the correctly detected regions to the tampered regions in ground truth.

Implementation Details

Our proposed method was implemented by Tensorflow and trained on NVIDIA GeForce RTX 3090 GPU. For the pre-training of the JPEG compression trace extractor, we cropped 200 TIFF images into 12800 48×48 patches as the training set. The batch size was set to 128, and the patches within a batch are randomly compressed with a quality factor $QF \in$[50, 100]. The Adam optimizer was used with the learning rate of 0.001, and the λ in Eq. 8 was set to 0.1.

In self-supervised training, each batch, sized 200, is divided into 50 groups (N=50), and each group is compressed with a different quality factor in [50, 100]. The Adam optimizer was used with a learning rate of 0.0001. Because the JPEG trace extractor already provides good features for contrastive classification, the training of our detection method can converge quickly. Since the output of the proposed method is gray images, for quantitative evaluation, we first inverted the gray value, then used gamma correction to highlight the tampered regions, and finally used the

Dataset	Pretrain		✓		✓
	SST			✓	✓
Spliced CoCo (SP)	S-S	14.23	17.35	14.19	**33.09**
	S-M	14.43	18.63	14.15	**33.49**
	M-M	13.59	18.94	13.89	**35.39**
Coverage (CM)	S-S	23.89	31.35	23.21	**55.75**
	S-M	23.62	32.11	23.17	**57.61**
	M-M	23.54	29.17	23.19	**56.46**
Korus (SP, CM, OR)	S-S	14.67	20.38	14.31	**65.47**
	S-M	13.75	19.96	13.32	**64.78**
	M-M	14.33	20.07	14.31	**60.72**

Table 3: The ablation study of the proposed method.

Otsu (Otsu 1979) method to binarize the entire image. Noted that the binarization will decrease the method's quantitative performance, especially the detection of small objects.

Ablation Study

To verify the feasibility and effectiveness of the proposed method, we made several variations based on the proposed method. In the first case, the parameters of the JPEG compression trace extractor were randomly initialized without pre-training and self-supervised training (Initialization). The second case is that the JPEG compression trace extractor was pre-trained without self-supervised training (Pre-train). Since the result of the pre-trained JPEG compression trace extractor is hard to locate the regions, we used the same enhancement method in Figure 4 to process the results. In the third case, the JPEG compression trace extractor was directly performed self-supervised training (SST). The last case is that the JPEG compression trace extractor was with pre-training and self-supervised training (Proposed).

In Table 3, We can see that the JPEG compression trace extractor in initialization and Self-supervised cases do not work at all. The reason is that the extractor is more likely to pay attention to the semantic information rather than the JPEG compression traces. Another side, the extractor with pre-training only can not complete the detection task well because the difference between the extracted compression traces is not strong enough to distinguish them under various local forgeries. Therefore, pre-training can make the network pay attention to the JPEG compression traces, and self-supervised training transfers the extracted compression traces to other spaces for strengthening their difference, which makes the proposed method can detect various local forgeries, even the tampered images are saved by JPEG compression many times (M-M).

Comparison with the State-of-the-Art

In comparison experiments, we selected two state-of-the-art local forgery detection methods in JPEG format: Park (Park et al. 2018) and CAT-Net (Kwon et al. 2021), which represent the vast majority of existing JPEG local forgery detection methods. Meanwhile, we choose two general forgery detection methods: Noiseprint (Cozzolino and Verdoliva 2019) and ManTra (Wu, AbdAlmageed, and Natarajan 2019). The comparison detection methods will be set according to their paper's best experimental configuration and training parameters. All detection methods will be evaluated across datasets.

Dataset		Method				
		Noise.	Mantra	Park	CAT.	Proposed
Spliced CoCo (SP)	S-S	22.62	**34.26**	✗	—	33.09
	S-M	16.39	32.08	30.54	—	**33.49**
	M-M	22.62	31.27	14.72	—	**35.39**
	Mean	20.42	32.54	22.63	—	**33.99**
Coverage (CM)	S-S	29.25	21.83	✗	14.34	**55.75**
	S-M	30.87	20.19	20.98	**74.91**	57.61
	M-M	32.74	20.29	16.45	13.36	**56.46**
	Mean	30.95	20.77	18.72	34.19	**56.61**
Korus (SP, CM, OR)	S-S	39.59	21.93	✗	15.42	**65.47**
	S-M	35.79	20.31	36.47	60.99	**64.78**
	M-M	35.84	20.91	13.19	12.55	**60.72**
	Mean	37.07	21.05	24.83	29.65	**63.66**

Table 4: The comparative experiments across different datasets. (✗ indicates illegal format and unable to work; — indicates that the method is trained on this dataset.)

Detection under Various Local Forgery. We carried out comparative experiments and showed the result in Table 4 and the left part of Figure 7. We can see that general local forgery detection methods cannot achieve good detection results in JPEG images. Mantra only showed better detection results on the Spliced CoCo dataset. The JPEG image forgery detection methods Park and CAT-Net can only play a role in the S-M situation. Compared with these methods, the proposed method can achieve stable detection results for three types of local forgery in different situations.

Detection under Various JPEG Compression Cases. In Table 4 and Figure 6, we can see Park and CAT-Net perform well in the S-M JPEG compression case because they are proposed only for the S-M case. However, in real world, the forgery case are far more complicated than those in the experiments. Since our method can achieve stable detection results for three types of local forgery in various JPEG com-

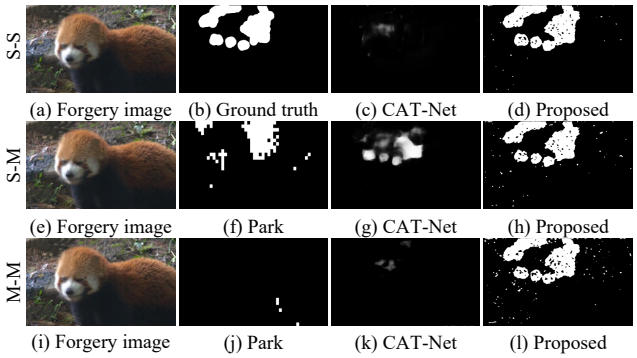


Figure 6: The experimental results of an object-removal forgery image under different situations.

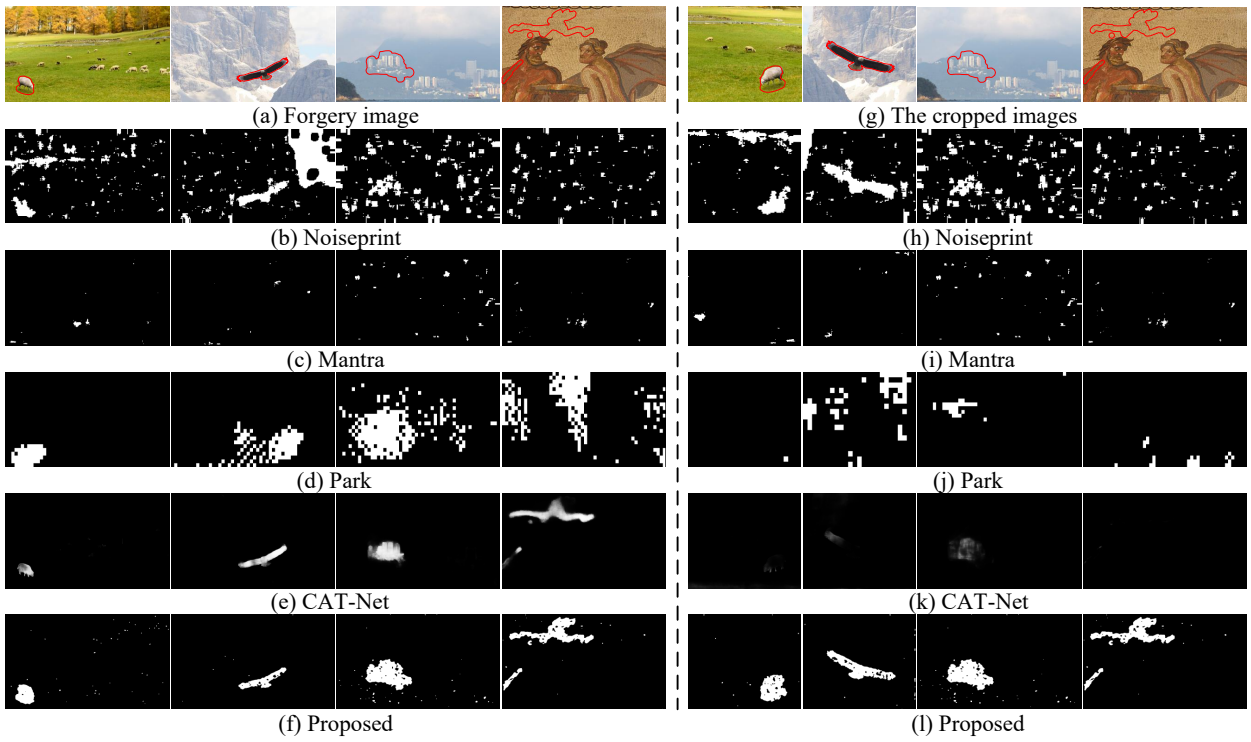


Figure 7: The examples of local forgery localization results. The first two columns are Splicing; the 3^{rd} and 4^{th} columns are Copy-move and Object removal, respectively; the 5^{th} and 6^{th} columns show the detection result after we cropped an area from the image and flipped it horizontally; the 7^{th} and 8^{th} columns are the detection result after we cropped just one row and one column from the top left corner.

Dataset		Method		
		Park	CAT-Net	Proposed
Spliced CoCo (SP)	S-S	✕	—	**32.31**
	S-M	14.18	—	**31.12**
	M-M	14.19	—	**34.61**
	Mean	14.19	—	**32.68**
Coverage (CM)	S-S	✕	17.66	**55.43**
	S-M	22.14	17.93	**57.45**
	M-M	21.19	15.76	**56.24**
	Mean	21.67	17.12	**56.37**
Korus (SP, CM, OR)	S-S	✕	11.5	**64.97**
	S-M	13.86	7.08	**60.99**
	M-M	13.69	7.69	**59.87**
	Mean	13.76	8.76	**61.94**

Table 5: The detection results under cropping attacks.

pression cases, we have confidence that it is more robust and practical in real application.

Robust Experiments

When we transmit an image over social networks, it is usually compressed and cropped, and we need to observe the JPEG compression traces after these operations for forensics. Therefore, we randomly cropped the test images and re-tested the detection methods. The quantitative results are recorded in Table 5, and the qualitative experimental examples are shown in the right part of Figure 7.

The results show the cropping attack is fatal to those methods based on block DCT coefficients. Although CAT-Net combines the RGB and the DCT stream, it cannot resist to cropping attacks. However, our method pays attention to different JPEG compression chains within the forged images, which is robust to cropping attacks. Therefore, the three types of local forgery can still be well detected in different situations.

Conclusions

Current double JPEG detection methods based on image block DCT coefficients only work in very limited situations and cannot be applied in the real world. To overcome their limits, we propose the concept of compression chains and distinguish compression chains of tampered and non-tampered regions to solve the local forgery detection in JPEG images. In the implementation, we design a compression trace extractor and enhance the discrimination ability of learned traces by using a self-supervised training method. The experiments demonstrate the effectiveness of the proposed method in many situations and prove its ability to resist cropping attacks.

Acknowledgements

This work was supported in part by the National Key Research and Development Project under Grant 2019YFE0110800, in part by the National Natural Science Foundation of China under Grant 62172067 and Grant 61976031, in part by the Natural Science Foundation of Chongqing for Distinguished Young Scholars under Grant CSTB2022NSCQ-JQX0001, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJQN202200635. Our deepest gratitude goes to the anonymous reviewers for their careful work and thoughtful suggestions that have helped improve this paper substantially.

References

Amerini, I.; Uricchio, T.; Ballan, L.; and Caldelli, R. 2017. Localization of JPEG double compression through multi-domain convolutional neural networks. In *2017 IEEE Conference on computer vision and pattern recognition workshops (CVPRW)*, 1865–1871. IEEE.

Bappy, J. H.; Roy-Chowdhury, A. K.; Bunk, J.; Nataraj, L.; and Manjunath, B. 2017. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE international conference on computer vision*, 4970–4979.

Bappy, J. H.; Simons, C.; Nataraj, L.; Manjunath, B.; and Roy-Chowdhury, A. K. 2019. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7): 3286–3300.

Bi, X.; Wei, Y.; Xiao, B.; and Li, W. 2019. RRU-Net: The ringed residual U-Net for image splicing forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Bianchi, T.; and Piva, A. 2011. Detection of nonaligned double JPEG compression based on integer periodicity maps. *IEEE transactions on Information Forensics and Security*, 7(2): 842–848.

Bianchi, T.; and Piva, A. 2012. Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3): 1003–1017.

Bilal, M.; Habib, H. A.; Mehmood, Z.; Yousaf, R. M.; Saba, T.; and Rehman, A. 2021. A robust technique for copy-move forgery detection from small and extremely smooth tampered regions based on the DHE-SURF features and mDB-SCAN clustering. *Australian Journal of Forensic Sciences*, 53(4): 459–482.

Bondi, L.; Lameri, S.; Guera, D.; Bestagini, P.; Delp, E. J.; and Tubaro, S. 2017. Tampering Detection and Localization Through Clustering of Camera-Based CNN Features. In *CVPR Workshops*, 1855–1864.

Bunk, J.; Bappy, J. H.; Mohammed, T. M.; Nataraj, L.; Flenner, A.; Manjunath, B.; Chandrasekaran, S.; Roy-Chowdhury, A. K.; and Peterson, L. 2017. Detection and localization of image forgeries using resampling features and deep learning. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, 1881–1889. IEEE. ISBN 1538607336.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Cozzolino, D.; and Verdoliva, L. 2019. Noiseprint: A CNN-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15: 144–159.

Deng, C.; Li, Z.; Gao, X.; and Tao, D. 2019. Deep multi-scale discriminative networks for double JPEG compression forensics. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2): 1–20.

Farid, H. 2009. Exposing digital forgeries from JPEG ghosts. *IEEE transactions on information forensics and security*, 4(1): 154–160.

Fu, D.; Shi, Y. Q.; and Su, W. 2007. A generalized Benford's law for JPEG coefficients and its applications in image forensics. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, 574–584. SPIE.

Korus, P.; and Huang, J. 2016. Evaluation of random field models in multi-modal unsupervised tampering localization. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6.

Kwon, M.-J.; Yu, I.-J.; Nam, S.-H.; and Lee, H.-K. 2021. CAT-net: Compression artifact tracing network for detection and localization of image splicing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 375–384.

Li, H.; and Huang, J. 2019. Localization of deep inpainting using high-pass fully convolutional network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8301–8310.

Li, H.; Luo, W.; and Huang, J. 2017. Localization of diffusion-based inpainting in digital images. *IEEE transactions on information forensics and security*, 12(12): 3050–3064.

Li, Y.; and Zhou, J. 2018. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Transactions on Information Forensics and Security*, 14(5): 1307–1322.

Liang, Z.; Yang, G.; Ding, X.; and Li, L. 2015. An efficient forgery detection algorithm for object removal by exemplar-based image inpainting. *Journal of Visual Communication and Image Representation*, 30: 75–85.

Lin, Z.; He, J.; Tang, X.; and Tang, C.-K. 2009. Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. *Pattern Recognition*, 42(11): 2492–2501.

Liu, B.; and Pun, C.-M. 2018. Deep fusion network for splicing forgery localization. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.

Mayer, O.; and Stamm, M. C. 2019. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 15: 1331–1346.

Niu, Y.; Tondi, B.; Zhao, Y.; Ni, R.; and Barni, M. 2021. Image Splicing Detection, Localization and Attribution via JPEG Primary Quantization Matrix Estimation and Clustering. *IEEE Transactions on Information Forensics and Security*, 16: 5397–5412.

Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1): 62–66.

Park, J.; Cho, D.; Ahn, W.; and Lee, H.-K. 2018. Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network. In *Proceedings of the European conference on computer vision (ECCV)*, 636–652.

Rao, Y.; and Ni, J. 2021. Self-supervised domain adaptation for forgery localization of JPEG compressed images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15034–15043.

Ruiz, H.; Chaumont, M.; Yedroudj, M.; Amara, A. O.; Comby, F.; and Subsol, G. 2021. Analysis of the scalability of a deep-learning network for steganography "Into the Wild". In *International Conference on Pattern Recognition*, 439–452. Springer.

Wang, Q.; and Zhang, R. 2016. Double JPEG compression forensics based on a convolutional neural network. *EURASIP Journal on Information Security*, 2016(1): 1–12.

Wang, W.; Dong, J.; and Tan, T. 2014. Exploring DCT coefficient quantization effects for local tampering detection. *IEEE Transactions on Information Forensics and Security*, 9(10): 1653–1666.

Wang, X.; Niu, S.; and Wang, H. 2021. Image inpainting detection based on multi-task deep learning network. *IETE Technical Review*, 38(1): 149–157.

Wen, B.; Zhu, Y.; Subramanian, R.; Ng, T.-T.; Shen, X.; and Winkler, S. 2016. COVERAGE — A novel database for copy-move forgery detection. In *2016 IEEE International Conference on Image Processing (ICIP)*, 161–165.

Wu, Y.; Abd-Almageed, W.; and Natarajan, P. 2018. Busternet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European conference on computer vision (ECCV)*, 168–184.

Wu, Y.; AbdAlmageed, W.; and Natarajan, P. 2019. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9543–9552.

Yang, J.; Zhang, Y.; Zhu, G.; and Kwong, S. 2020. A clustering-based framework for improving the performance of JPEG quantization step estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(4): 1661–1672.

Zhang, D.; Liang, Z.; Yang, G.; Li, Q.; Li, L.; and Sun, X. 2018. A robust forgery detection algorithm for object removal by exemplar-based image inpainting. *Multimedia Tools and Applications*, 77(10): 11823–11842.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7): 3142–3155.

Zhang, Y.; Song, W.; Wu, F.; Han, H.; and Zhang, L. 2020. Revealing the traces of nonaligned double JPEG compression in digital images. *Optik*, 204: 164196.

Zhong, J.-L.; and Pun, C.-M. 2019. An end-to-end dense-inceptionnet for image copy-move forgery detection. *IEEE Transactions on Information Forensics and Security*, 15: 2134–2146.

Zhou, P.; Han, X.; Morariu, V. I.; and Davis, L. S. 2018. Learning rich features for image manipulation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1053–1061.

Zhu, Y.; Chen, C.; Yan, G.; Guo, Y.; and Dong, Y. 2020. AR-Net: Adaptive attention and residual refinement network for copy-move forgery detection. *IEEE Transactions on Industrial Informatics*, 16(10): 6714–6723.