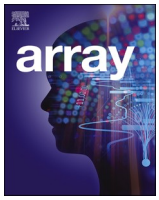
Array 18 (2023) 100279

Contents lists available at ScienceDirect

Array

journal homepage: www.sciencedirect.com/journal/array



IMG_{CAT}: An approach to dismantle the anonymity of a source camera using correlative features and an integrated 1D convolutional neural network

Muhammad Irshad [a], Ngai-Fong Law [a,*], K.H. Loo [a], Sami Haider [b]

[a] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, 11 Yuk Choi Rd., Hung Hom, Hong Kong
[b] Glasgow College University of Engineering Science and Technology Chengdu, China

ARTICLE INFO

Keywords:
1D CNN
Source camera identification
Feature extractions
Seam carving
Computer vision
Image processing

ABSTRACT

With the proliferation of smartphones, digital data collection has become trivial. The ability to analyze images has increased, but source authentication has stagnated. Editing and tampering of images has become more common with advancements in signal processing technology. Recent developments have introduced the use of seam carving (insertion and deletion) techniques to disguise the identity of the camera, specifically in the child pornography market. In this article, we focus on the available features in the image based on PRNU (photo response nonuniformity). The forced-seam sculpting technique is a well-known method to create occlusion for camera attribution by injecting seams into each 50 × 50 pixel block. To counter this, we perform camera identification using a 1D CNN integrated with feature extractions on 20 × 20 pixel blocks. We achieve state-of-the-art performance for our proposed IMG_{CAT} (image categorization) in three-class classification over the baselines (original, seam removed, seam inserted). Based on our experimental findings, our model is robust when dealing with blind facts related to the questionable camera.

1. Introduction and related work

A picture is worth a thousand words. Photos do not lie. These banal phrases have been around for a long time, and there has been little reason to question the reliability of a photograph in the current era until the recent presence of powerful graphics editors and developments in digital media technology [1]. Many low-cost software are accessible and make it trivial to manipulate images (e.g., Adobe Photoshop, Corel Paint Shop Pro, Skylum, Canva, Stencil, etc.), especially digital images.

Image attribution is a vital job in the framework of digital multimedia forensics [2]. In recent years, it has added a great deal of consideration to combat counterfeiting and exposing the truth. Modern cameras (DSLR, compact, and cell phone models) generate large images with millions of pixels, which offer sufficient evidence for reliable device identification [3]. In the court of law, captured images or video materials are presented as substantial proof. Therefore, tempered images or videos may be shown as false evidence [4]. Due to its implications in several fields of application, the identification of camera models is one of the problems in the image forensics community [5]. To do this, the brand and model of the camera that is used to take an image must be determined. Solving this problem could help an expert point out the possessor of illicit and controversial material (e.g., pornographic shots, terrorist act scenes, etc.). Innocent peoples are sometimes trapped by exploiting intrinsic artifacts in visual proof provided by the honorable courts. In the Washington Post, "Democracy dies in the darkness," a famous Boston Marathon bombing investigation firm claimed a piece of serious evidence tampering with a group of activists accused of plotting to overthrow the government [6].

Seam deletion and seam insertion resize the image with better flexibility than cropping and resampling and operate on a simple algorithm [7]. Here, we analyze artifacts caused by seam carving using only retargeted images because the positions where the seam is inserted and removed vary with the content and region in the image [8,9]. Accordingly, retargeted images generated by seam-carving are more challenging to classify than linearly scaled images with periodic characteristics, since the seams are globally scattered according to the intrinsic features of an image [10]. The seam can horizontally or vertically cross the image, i.e., from left to right or top to bottom, and each pixel in the seam has 8 connections [11]. In seam carving, the "vital features" in an image are left unaffected when the image is resized, and it is typically presumed that the "vital features" are not categorized by the low-energy pixels [7,12,13]. Thus, seam carving has become one of

* Corresponding author.
E-mail addresses: mirsha@polyu.edu.hk (M. Irshad), ngai.fong.law@polyu.edu.hk (N.-F. Law), Kh.loo@polyu.edu.hk (K.H. Loo), samiahmed.haider@glasgow.ac.uk (S. Haider).

https://doi.org/10.1016/j.array.2023.100279
Received 12 November 2022; Received in revised form 18 January 2023; Accepted 20 January 2023
Available online 8 February 2023

the most popular image resizing algorithms and is included as a feature in popular image editing software, such as Photoshop CS6 and GIMP. Traditional resizing techniques (e.g., center cropping, interpolation, linear scaling) adjust the image fit for the target size. However, the vital image content may be spoiled, as shown in Fig. 1.

Fig. 1 shows different methods for image resizing. In general, resizing is used to overcome incompatibilities between modules since each device and application provides a different size and aspect ratio [14,15]. For instance, as shown in Fig. 1(b), (c), and (d), when traditional image resizing methods are applied at an equal rate along one axis, the aspect ratio can be changed; it does not make any significant difference in contrast, since image (e) has the same amount of magnitudes. We noticed some startling points. As mentioned, seams are passed through low-energy regions and preserve important contents such as whether we see the light board and paper sheet. The seams passed through characters such as "B I R" and "EE S″ on the light board, and they concatenated "HBD" on the paper sheet accordingly. Additionally, the important content of the image is more prominent than the remainder of the images. Finally, the seam insertion image in Fig. 1(f) clearly defines the extended area along the x- and y-axes. Additionally, since seam carving is a common malware attack method, identifying seam-removed and seam-inserted traces has become a top priority in image forensics [16]. Using this retargeting approach, the original content and objects can be deliberately removed in court [17]. As mentioned in Ref. [8], it prevents the synchronization of invisible watermarks and produces abnormal watermark extraction. Thus, classifying these forgeries has become an important issue.

1.1. Application of artificial intelligence in forensic studies

In image forensics, image manipulation causes changes in intrinsic statistics [18,19], and forensic techniques to capture seam-carving artifacts must look at the characteristics of the retargeting process. The statistical changes caused by seam removal or insertion in retargeted images can be captured by handcrafted feature-based approaches [20,21]. Sarkar et al. proposed a set of 324-dimensional Markov features (Shi-324) and used a supervised support vector machine framework [22]. Wei et al. introduced an SVM-based approach in Ref. [23] using three types of patch transition probability matrices and 2×2 blocks. A set of half-seam features, energy features, and noise-based features from the local binary pattern domain was combined in Hung et al. [24] to highlight the local texture artifacts. According to Ref. [25], a set of directional derivatives and Gabor residuals generally performed well for forensic tasks. These conventional approaches have performed well but do not fully satisfy forensic detection needs, since forensic traces can be lost while handcrafted features are generated [26]. There are also instances when two independent algorithms are required to detect the seam insertion and removal [27]. It takes two tests to authenticate an image, which results in a high false-alarm rate.

To address the implicit problem of these feature-based approaches, techniques using different learning frameworks have been proposed to enable the network to automatically learn forensic features. Many approaches to CNN-based multimedia forensics have been proposed using deep learning [8,19,20]. CNNs for computer vision can learn features from the data using preprocessing layers or network components that are specialized for learning low-level features. CNN-based forensic approaches have been designed to learn forensic features while suppressing the content of the image. Bayar et al. [28] introduced a constrained convolution layer, which is referred to as BayarNet, that forces the CNN model to learn prediction error filters to produce low-level forensic features. According to Ref. [29], H-VGG is used in conjunction with VGGNet [30] with high-pass filtering (HPF). This model suggests double compression artifacts within a decoded intracoded frame (I-frame) of H.264 video. Rössler et al. [31] created a Face Forensics dataset consisting of fake videos and showed that Xception could detect artifacts during the generation of fake faces. A framework for steganalysis in both spatial and JPEG domains was presented by Boroumand et al. [25]. To explore low-level artifacts, SRNet operates without the pooling layer in the early and middle stages. As a CNN-based method, Ye et al. proposed YeNet [32], which includes a preprocessing layer for seam-carved image detection.

The paper proposes an indirect correlation analysis method to identify the source camera of a seam-carved (insertion, deletion) image. Since noise residues and PRNU behave differently in the matched and unmatched cases, the proposed method uses the correlation pattern between them. Matching refers to taking the test image from the corresponding camera. From these patterns, five types of features are extracted and fed into the CNN network to identify the source camera. The proposed technique selects suitable blocks from the test image for source identification. Previous research has shown successful source identification considering 50×50 uncarved blocks from multiple seam-carved images [33–35] using the force seam carved method. The theoretical and novel contributions of the proposed method in comparison with the others are highlighted below.

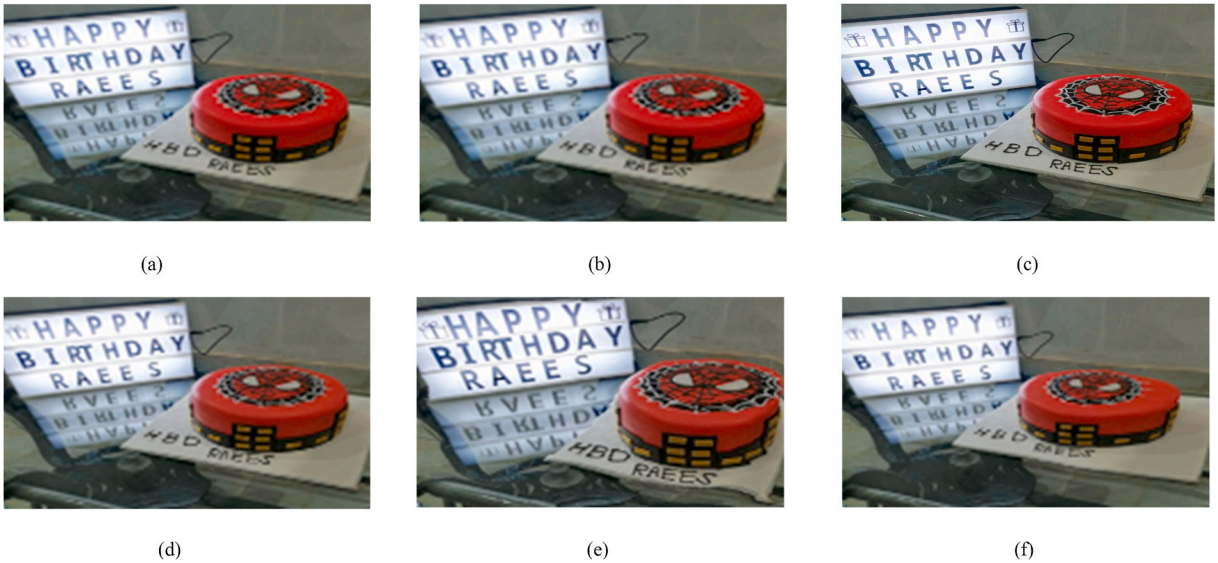


Fig. 1. (a) Original image. Resized versions of (a) using (b) scaling, (c) cropping, and (d) cubic interpolation. (e) Seam deletion and (f)) seam insertion images. The image sizes of (a) and (f) are 1241×1240 and 1441×1440, respectively, and 941×940 for all other resized images.

- What is the purpose of the proposed technique? The proposed technique is a method for source identification in digital images, with a focus on accurately identifying the source camera even when there is only a small amount of uncarved data available.
- How does the proposed technique achieve source identification? The proposed technique uses a data-driven algorithm called IMG_{CAT}, which is based on 1D convolutional neural networks (CNNs) and is designed to learn the features characteristic of each camera device.
- How does the proposed technique compare to previous methods? Previous research has shown successful source identification using 50 × 50 uncarved blocks from multiple seam-carved images using the force seam carved method. In contrast, the proposed method can accurately identify the source camera even when there is only one uncarved block as small as 20 × 20 on an image. Additionally, the proposed method has been shown to be superior to state-of-the-art works in terms of both seam insertion and deletion, and has demonstrated robustness against unknown patterns in extensive experiments.
- What are some key features of the proposed technique? Some key features of the proposed technique include its ability to accurately identify the source camera even when there is only a small amount of uncarved data available, its use of 1D CNNs for the IMG_{CAT} algorithm, and its ability to classify images based on three classes (original, seam removal, and seam insertion).

The remainder of the paper is organized as follows: In Section 2, we examine two major methodologies, i.e., seam insertion and deletion. Section 3 deals with acquiring forged images, while Section 4 examines the selection of blocks and numerical analyses experimental approach and decision matrices calculations, correlation patterns, features. We discuss the impact of a 1D CNN model on our proposed approach in Section 5, followed by a conclusion in Section 6.

2. Review on seam removal and seam insertion

Seam carving (SC) resizes an image without disturbing important image structures by characterizing the image structure using a gradient magnitude [36]. In particular, highly textured regions with large variations in pixel intensities and large gradient magnitudes are not resized, while smooth regions with small gradient magnitudes are chosen for removal. Thus, visual artifacts arising from the scaling operations can be minimized. Mathematically, the absolute gradient magnitude $e(I)$ of an $n_x \times m_y$ I is defined as

$$e(I) = \left|\frac{\partial}{\partial x} I\right| + \left|\frac{\partial}{\partial y} I\right| \tag{1}$$

where I represents an image, $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ denote respectively the partial derivatives of the image with respect to the x and y directions, and $|.|$ represents the absolute value. A large value of $e(I)$ represents highly textured areas that should not be scaled, whereas a small value of $e(I)$ indicates smooth regions that can be scaled with no obvious visual distortions.

Seams are lines joining pixels with the smallest variation, i.e., lines with the smallest sum of the absolute values of the gradient magnitudes in an image. These lines are removed to scale down the size of an image. They can also be copied and inserted to scale up the size of an image [12]. Let x be a mapping that indicates the relationship between the row index and the y coordinate of the seam, i.e., $x : [1,\ldots,n] \rightarrow [1,\ldots,m]$. The vertical seam is defined as

$$s^x = \left\{s_i^x\right\}_{i=1}^n = \{(i, x(i))\}_{i=1}^n$$

such that

$$|x(i) - x(i-1)| \leq 1 \forall i \tag{2}$$

Removing a vertical seam from an image decreases its width by one pixel, while removing a horizontal seam decreases its height by one pixel. The set of pixels in the vertical seam can be written as

$$I_x = \left\{I\left(s_i^x\right)\right\}_{i=1}^n = \{I(i, x(i))\}_{i=1}^n \tag{3}$$

To identify vertical seams, the cost can be defined using the gradient magnitudes as follows:

$$E(I_x) = \sum_{i=1}^{n} e\left(I\left(s_i^x\right)\right) \tag{4}$$

The optimal seam s^* minimizes the cost, i.e.,

$$s^* = \min_s E(I_x) = \min_s \sum_{i=1}^{n} e\left(I\left(s_i^x\right)\right) \tag{5}$$

Similarly, let y be a mapping that indicates the relationship between the column index and the x coordinate of the seam, i.e., $y : [1,\ldots,m] \rightarrow [1,\ldots,n]$. The horizontal seam is defined as

$$s^y = \left\{s_j^y\right\}_{j=1}^m = \{(y(j), j)\}_{j=1}^m$$

such that

$$|y(j) - y(j-1)| \leq 1 \forall j \tag{6}$$

Then, the set of pixels in the horizontal seam is

$$I_y = \left\{I\left(s_j^y\right)\right\}_{j=1}^m = \{I(y(j), j)\}_{j=1}^m \tag{7}$$

Similar to the vertical seam, the horizontal seam is obtained by minimizing the cost, i.e.,

$$s^* = \min_s E\left(I_y\right) = \min_s \sum_{j=1}^{m} e\left(I\left(s_j^y\right)\right) \tag{8}$$

In Fig. 2, the seam energy functions are illustrated by the way low-energy pixels are removed during seam carving, while relevant objects (high energy pixels) are preserved.

The optimal seam can be found using dynamic programming [37]. In other words, the image is traversed from the second row to the last row by searching the cumulative minimum absolute gradient value among all possible connected paths. Let M be the cumulative minimum absolute gradient value for pixel (i,j) in image I. It can be expressed as

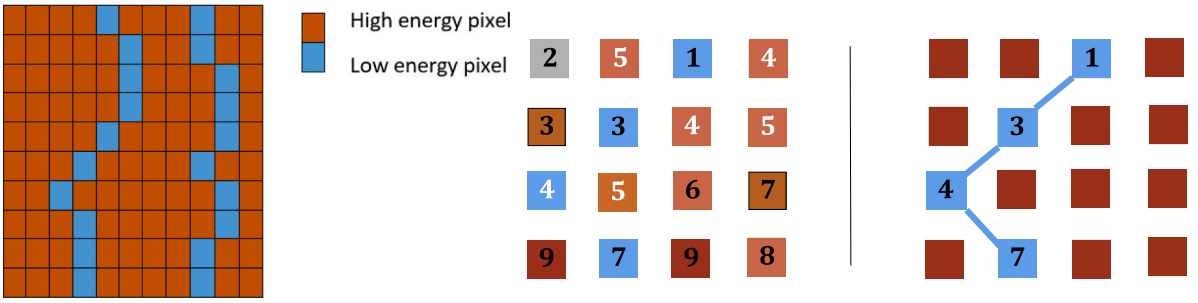
$$M = \cdots + e(i-1, j) + \min\begin{pmatrix} e(i, j-1), \\ e(i, j), e(i, j+1) \end{pmatrix} + \cdots \tag{9}$$

M is the optimal vertical seam location by its minimum value in the last row. The path corresponding to the minimum value of M refers to the set of points that constitute the seam. As a result, we can backtrack from this minimum entry to find the other points in the optimal seam. Moreover, there is no difference in the seam selection process for seam carving and seam insertion [38]. There are improved techniques in seam insertion; conventionally, three pixels in a row are chosen. The fourth pixel is inserted as shown in Fig. 3.

In Fig. 3, the distance between z and x is larger than that between z_1 and x; similarly, the distance between z and y is greater than that between z_2 and y, which can be explained by Equation (10).

$$z_1 = round\left(\frac{x+z}{2}\right) z_2 = round\left(\frac{z+y}{2}\right) \tag{10}$$

The original three pixels are x, z and y. When one pixel is added, the new four pixels are x, z_1, z_2 and y. Only one new pixel value is introduced when the selected seam lies along the border. After the seam insertion, for example, when [x, z] is replaced by [a, z_1, z] and the border pixel (or the pixel through which the seam passes) is x (or z), the new value of the border pixel (z_1) is equal to $round\left(\frac{x+z}{2}\right)$.



(a). Energy between pixels (b). energy map (c). cumulated energy map M(i,j)

Fig. 2. Features for removing seams.

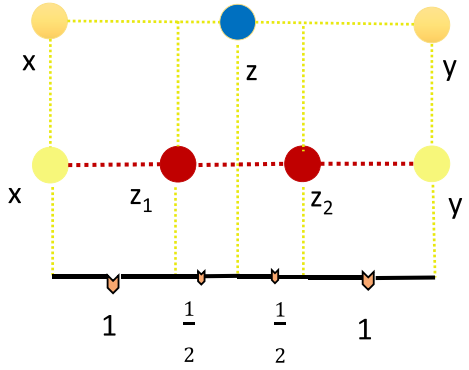


Fig. 3. Location of pixels; blue is before and red is after the insertion. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

3. Acquisition of forged images

The first step is to generate forged images using restricted force seam carving, which is a subtype of forced seam carving described in Ref. [35]. In forced seam carving, which is also known as relaxed seam carving, according to Refs. [9,34], the entire image is split into N × N blocks, i.e., 50 × 50 blocks. In this case, the condition must be met by passing the seam from each split block so that no block remains uncarved. Unlike conventional seam carving, forced seam carving does not rely on energy or random pixel selection. The further development of forced seam carving results in a technique called restricted seam carving (removal only), where the author restricts the block selection by 25 × 25 blocks [35]. Using the former technique, the author successfully used a series of test images to make a camera attribution. According to our approach, we split the blocks following that approach by 20 × 20 while simultaneously considering both aspects (insertion and deletion). In the field of image forensics for source attribution, this aspect of the problem has not been investigated in any research studies. This concept is illustrated in Fig. 4, which shows an example of relaxed and restricted seam carvings. For comparison, two images were taken from Microsoft Lumia 640 LTE and Apple iPad Mini.

In the left-hand side, the Microsoft Lumia 640 LTE (Landscape) camera figures correspond to the relaxed seam carving method. Relaxed seam carving is a fairly fragile technique where most of the carved blocks are smaller than 50 × 50. Conversely, a restricted seam carving right-hand side Apple_iPadMini (Portrait) camera makes every block smaller than the dimensions of 20 × 20. To anonymize the identification, the latter technique proves to be more effective than the former.

4. Proposed approach

This section is primarily intended to present an explanation of the computational preprocessing operation which is applied before the CNN architecture based on five features extracted from the test images. Next, we examine CNN architecture and design considerations in more detail. Finally, we discuss some key parameters that are selected or adjusted during the training and testing processes. The framework of the proposed algorithm IMG_{CAT} is illustrated in Fig. 5.

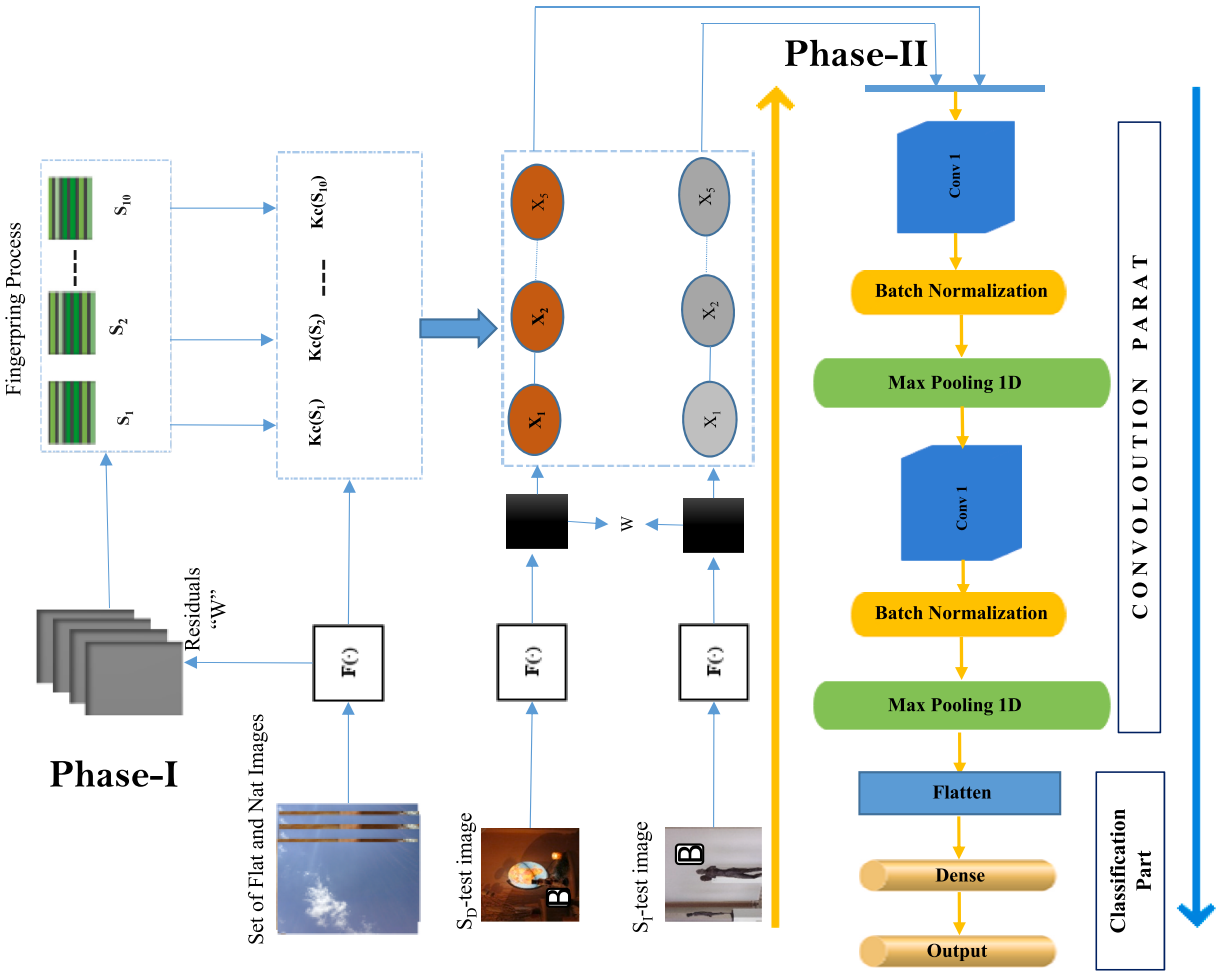
4.1. Phase I

4.1.1. Fingerprinting process for cameras

Photo Response Non-Uniformity is a pattern of multiplicative noise arising from different sizes of imaging sensor cells. Because of manufacturing errors, sensor cells are not precisely uniform and generate pixels with slight differences in luminance when exposed to the same light intensity. As a result, for an image I generated by a given camera, a simplified multiplicative model is as follows:

$$I = I^{(0)} + I^{(0)}K + \Theta \tag{11}$$

Fig. 4. Example drawing images for relaxed and restricted seam carvings.



S_D = A test image with seam deletion applied, S_I = A test image with seam insertion applied, B = A block within image, F(.) = standard Denoising algorithm e.g BM3D

Fig. 5. Overview of the forensic approach algorithm IMG_{CAT}.

where $I^{(0)}$ is noiseless image, K is the PRNU pattern, Θ is a condensated independent components of random noise. All images acquired by each device contain the PRNU, which is unique, stable, and present in all of the images. In this way, it is considered a legitimate device fingerprint [39].

To test the efficacy of the proposed algorithm, two phases of experiments are conducted. In the first phase of the experiment, the set $\{C_l \in C, l = 1, ..., L\}$ contains all cameras under testing, where L is the total number of cameras, i.e., 11. Using different combinations of flat-field and natural images, 250 in total, as shown in Table 2, the fingerprint of the camera is obtained. Assume that we have M images taken by camera C. These images are randomly divided into S groups (in our case there are 10 groups with 25 randomly selected images in each group), i.e., $\{S_1, S_2, \ldots \ldots S_{10}\}$. In the offline camera fingerprint construction step, a set of camera fingerprints can be estimated from M images taken by the camera as follows: $\{K_{C_l}(S_1), K_{C_l}(S_2), \cdots, K_{C_l}(S_{10})\}$. Based on a camera device c with a PRNU fingerprint K c and a query images $I(S_D)$ seam deletion and $I(S_I)$ seam insertion, a binary hypothesis testing problem is defined as

H0. $I(S_D)$ and $I(S_I)$ was not taken with camera c, therefore does not contain K_C.

H1. $I(S_D)$ and $I(S_I)$ was taken with camera c, therefore contain K_C to determine if the query images were shot with that camera.

In order to detect such matching, a cross-correlation test can be used, for example in seam deletion case:

$$\rho\left(\acute{K}_{c,} W^{(SD)}\right) = \sum_{i=1}^{h} \sum_{j=1}^{w} \acute{K}_c(i,j).W^{(SD)}(i,j) \tag{12}$$

where W is residual from filter. When $\rho(\acute{K}_{c,} W^{(SD)}) > \tau$ then H_1 is proved. As a result, the query image is attributed to camera C. The threshold τ is

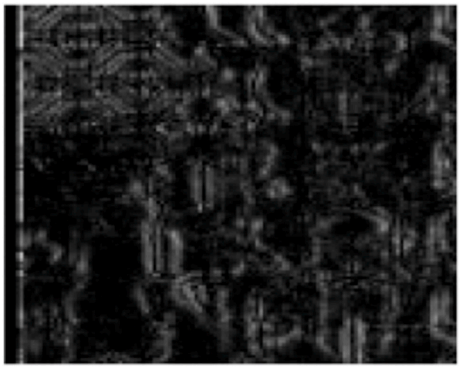


Fig. 6. An illustrative Block B with a size of 128 × 128 with the maximum value of $E_{X,i}$

properly set so that a desired target value can be used to limit the false alarm probability.

4.1.2. Dual-gradient energy function-based block selection

Within the testing images (constructed by following the steps in Sections 2 and 3), the proposed method chooses a block that has not been subjected to seam removal or addition. The seams seldom run across a highly textured area or regions with edges. Therefore, these regions are less likely to be infected. A gradient magnitude can help identify these uncarved blocks. Let $E_{X,i}$ be the 128 × 128 of the gradient of a block of the image X. By shifting the block within the testing photo, the maximum value of $E_{X,i}$ among all blocks is found. Let P be the block index corresponding to the maximum value of $E_{X,i}$ and N' be the total number of blocks. Fig. 6 illustrates an example of block "B", which fulfills the requirement.

4.1.3. Feature extraction

As discussed above, an uncarved block (P) is extracted from the seam carved image. The size of P is much smaller than that of the camera fingerprint, and a search will be performed within the searching window in the camera fingerprint to find the potential correspondence position between P and the camera fingerprint. We define the maximum correlation coefficient as $corr_{j,c_l}$ and the matched position $(x^*, y^*)_{j,c_l}$ as the location that maximizes $corr_{j,c_l}$, where j is the location of the block within a test image of camera C, i.e.,

$$Corr_{j,c1} = \max corr\left(A_{j,c1}(x,y) \otimes f(I_P), R_P\right) \tag{13}$$

$$(x^*, y^*) = \text{argmax} corr\left(A_{j,c1}(x,y) \otimes f(I_P), R_P\right) \tag{14}$$

where argmax is the value of the block index with the maximum gradient magnitude. $A_{j,c_l}(x,y)$ is a sub block inside $K_{C_l}(jm)$ centered at (x, y). The size of $A_{j,c_l}(x,y)$ is identical to that of R_P.

Having obtained a set of indirect correlations $\{corr_{j,c_l}, j = 1, \ldots, r\}$ and matched positions $\{(x^*, y^*)_{j,c_l}, j = 1, \ldots, r\}$ for $\{C_l \in C, l = 1, \ldots, L\}$, features are extracted from them to devise a robust decision metric. The decision metric will be used to determine which camera device in $\{C_l \in C, l = 1, \ldots, L\}$ was used to capture the testing photo that has been seam-carved. For each camera device, five types of features are extracted. They are:

(1) The sum of indirect correlation from all camera fingerprints of camera C_l, i.e.,

$$x_{1,c_l} = \sum_{j=1}^{r} corr_{j,c_l} \tag{15}$$

(2) The change in indirect correlation

$$x_{2,c_l} = corr_{r,c_l} - corr_{1,c_l} \tag{16}$$

(3) The absolute difference of the change of the location $(x^*, y^*)_{j,c_l}$ with respect to the number of images to construct the camera PRNU, i.e.,

$$x_{3,c_l} = \sum_{j=1}^{r-1} \left| (x^*, y^*)_{j+1,c_l} - (x^*, y^*)_{j,c_l} \right| \tag{17}$$

(4) Let M be the 20 × 20 sub block of $corr(A_{j,c_l}(x,y) \otimes f(I_P), R_P)$ centered at $(x^*, y^*)_j$ and

$$\beta_j = \sum_{x=1}^{20} \sum_{y=1}^{20} \frac{\left(corr_{j,c_l} - M(x,y)\right)^2}{\left(corr_{j,c_l}\right)} \tag{18}$$

β_j tries to find the difference between the maximum correlation value $corr_{j,c_l}$ and the correlation values of its 20 × 20 neighbors. Then, the feature is obtained as $x_{4,c_l} = \sum_{j=1}^{r} \beta_j$

$$x_{5,c_l} = \beta_r - \beta_1 \tag{19}$$

These five features attempt to characterize the difference in correlation patterns and the change in matched locations.

4.2. Phase II

4.2.1. 1D convolutional neural network (CNN)

CNNs are deep learning methods that incorporate artificial neural networks to extract features from input [40]. For a 1D classification problem, a 1D CNN can be considered a mapping function $\mathscr{F}_\ominus = \mathbb{R}^{n \times c} \rightarrow \mathbb{R}^{m}$ that maps an input $x \in \mathbb{R}^{n \times c}$ to an output $\hat{y} \in \mathbb{R}^{m}$ based on the calculated parameters $\ominus$, where n is the input vector length, c is the input channel count, and m is the number of classes. Our problem can be formulated as follows: x is an input feature taken from a testing image that must be classified by $\mathscr{F}_\ominus$ as a vector of probabilities. The greatest output is arg $\max_{i \in \{1..5\}} \hat{y}_i$. A CNN consists of L hidden layers. In each layer l, $l \in \{1 \ldots. L\}$, contains n_l neurons with activated output $a^{(l)}$. A vector is constructed by computing the values of each neuron of the layer in a feed-forward propagation manner as follows:

$$a^{(l)} = g^{(l)}\left(w^{(l)}\, a^{(l-1)} + b^{(l)}\right) \forall l \in \{1 \ldots. L\} \tag{20}$$

where $g^{(l)}: \mathbb{R}^{n} \rightarrow \mathbb{R}^{n}$ is a nonlinear of layer l that ensures that only crucial neurons are activated (i.e., >0) and passes its output to the next layer. $w^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ are the weights, and $b^{(l)} \in \mathbb{R}^{n_l}$ are the biases learned in the training process. A function can be calculated from the output of layer L, i.e., the last hidden layer, as shown in the following equation:

$$\hat{y}i = a^{(L)} = g^{L}\left(w^{(L)}\, a^{L-1} + b^{L}\right) \tag{21}$$

With the output y_i obtained by performing forward propagation, the difference between the ground truth label y_i and predicted $\hat{y}_i$ is calculated as the loss $E_i = (y_i - \hat{y}_i)^2$, assuming squared loss. Then, a back-propagation method is used to calculate the contribution of every w toward this loss. It is calculated by computing the partial derivatives of loss with respect to each individual weight. Here is an example of the calculation with a single weight $w_3^{(L)}$, which corresponds to node 2 in Layer L- 1 to node 1 in Layer L as follows:

$$\frac{\partial E_i}{\partial w_3^{(L)}} = \left(\frac{\partial E_i}{\partial a_1^{(L)}}\right)\left(\frac{\partial a_1^{(L)}}{\partial z_1^{(L)}}\right)\left(\frac{\partial z_1^{(L)}}{\partial w_3^{L}}\right) \tag{22}$$

$$= 2\left(a_1^{(L)} - \mathscr{Y}_i\right)\left(g^{(L)\prime}\, z_1^{(L)}\right)\left(a_2^{(L-1)}\right) \tag{23}$$

where $g^{(L)}$ is the activation function of layer L, and $z^{(L)}$ is the input for that neuron.

4.2.2. Network architecture

In this phase, the internal weights of the model were reorganized over various iterations. Finally, the dataset features were used to train the model. In this part, the experiments for the proposed CNN architecture were implemented using the jupyter notebook of the Anaconda distribution. Compared to the commonly used neural network (NN), which takes vectors as the inputs and performs global feature extraction, 1D CNNs operate in a multi-scale manner from local to global and investigate the mutual information between locally available features. The input layer is denoted by N × 5 × 1, where N is the length of the input vector, e.g., N = 11., which is followed by dense layer to map the neurons; avoid the potential of overfitting and uses the rectified linear unit (ReLU) as an activation function. The layer has one direct connection to the previous layer and one direct link to the following layers. The consecutive layer is a MaxPooling1D, which is a representative pooling layer, employed to reduce the dimensionality of the

Table 1
The CNN layers for generated feature set.

Layer	Output shape	Kernel size
Conv1D	(5,32)	32
Batch normalization	(5,32)	32
MaxPooling1D	(3,32)	32
Conv1D	(3,16)	16
Batch normalization	(3,16)	16
MaxPooling1D	(3,16)	16
Flatten	(32)	–
Dense	(32)	32
Dense	(11)	11

feature maps. Unlike the approach of [28], where a 1 × 1 Conv layer is placed on a deeper layer, we sequentially place a 1 × 1 Conv layer that learns the association between the feature maps in deep layers. Additionally, a 1D CNN is used to reduce space requirements and speed up CPU time by extracting correlational values and other spatial features as input. Using a 1D CNN, we can extract short-term frequency domain features from time-domain signals. A dropout layer is combined with a convolution layer of 1D. Input matrix M is a normalized spatial representation with size (5,1). Convolution in 1D is computed as follows.

$$f = \sigma\left(\sum w_{h,k}.X + b\right) \tag{24}$$

Here, (h, k) represents the number of convolutional filters, and b represents the bias value that needs to be learned. σ is the nonlinear activation function of M matrix. Weight (w) and X are dot products (.) along with the bias value is used to compute output (f). In the convolution operation, as M is larger than filters, a sliding window is required. Using a stride of step size 1, we can extract local features that contain useful information for detecting features. Its characteristic is translational invariance, so the output can move in any direction. To reduce the number of feature neurons fed to the next layer, we apply the dropout layer. The loss function is meant to calculate the quantity that minimized during training. Categorical cross-entropy is a loss function that is an optimization function used to classify the data by predicting the probability of whether the data belongs to one class or the other. This loss function when used for multi-class classification provides an output that is one-hot encoded in the form of 0's and 1's.

Based on Table 1, we classified our data using a 5-layered architecture model. First, we have Conv1D, which has an activation function of "relu" and an input size of (N, 5). The filter size is 32, with a kernel size of 2. Conv1D follows a dense layer with a size of 16 and a similar activation function. Maxpooling1D's maximum value over a spatial window size-2 is down-sampled from the previous layer. In the end, we get the following output:

$$output = \frac{input - poolsize + 1}{stride} \tag{25}$$

Arrays of 2D values are flattened to 1D without affecting batch size. Normalization is done between layers of a neural network using batch normalization. When data are normalized, they are reduced in covariate shift by definition, and covariate shift is the change in data distribution. As a result, the internal layers of CNN tend to receive different input distributions. Additionally, it has a regularization effect on the data as shown in the following equation.

$$x = \left(\frac{x - mean}{standard_{deviation}}\right) * alpha + beta \tag{26}$$

where alpha and beta are learnable parameters. The proposed network, which consists of a combination of unique characteristics and purposes, automatically explores the forensic features in an end-to-end fashion.

4.3. Dataset construction

Currently, there is no open image database specifically designed for seam-carved images. We propose a dataset that includes models of DSLR cameras, compact cameras, and mobile phones. To conduct the experiment, we built a test image database from scratch. As a first step, two image databases are chosen: VISION and Dresden [41,42]. Cell phones dominate the vision dataset, so we chose to analyze cells from that dataset, while DSLR cameras from Dresden are used in our dataset. Table 2 summarizes the main features of the dataset. For each device, its camera model, its label, number of images collected, and their corresponding resolution are shown. In modern society, cell phones, compact cameras, and DSLR cameras are widely used, which makes them the perfect dataset to use.

A set of experiments and analyses was conducted to assess the performance of our algorithm IMG_{CAT}. For this purpose, 11 camera devices from two image databases (VISION and Dresden) were chosen, as shown in Table 2. For each camera device, 250 images were randomly selected as the training images. These 250 images were further divided into 10 groups with 25 images in each group. A set of camera fingerprints $\{K_C(m), K_C(2m), \cdots, K_C(rm)\}$ was obtained, where $r = 25$ and $m = 10$ for camera C.

4.4. An illustrative camera case

As an example, Fig. 7 was taken from Microsoft Lumia 640 LTE and Apple iPad Mini. Following the restricted seam carving method (forced seam carving), the images were processed for both seam insertion and deletion. In this process, the image was divided into $\frac{n}{2} \times \frac{n}{2}$ blocks, and at least one seam was removed or added to each block. It is imperative to understand that the images are carved to avoid being distorted and to conceal any information about tempering. It mimics the technique that content creators use to conceal the source of camera images. This experiment was conducted on a Windows 10 system using 32G DDR3, Intel® UHD graphic card 630 with 1G memory, MATLAB R2021a,

Table 2
List of camera devices, including their brand, model, image resolution, number of flat images (#Flat), number of natural images (#Nat), file format and total number of images.

S.No	Camera Model	Labeled	Image Resolution	#Flat	#Nat	Fil File Format	Total No of Images
1	Apple_iPadMini	C_1	2592 × 1936	100	150	JPEG	250
2	Canon_PowerShotA640_0	C_2	2592 × 1944	50	200	JPEG	250
3	Sony_DSC-T77_1	C_3	3648 × 2763	100	150	JPEG	250
4	Xiaomi_RedmiNote3	C_4	4608 × 2592	150	100	JPEG	250
5	Nikon_D200_0	C_5	4352 × 3264	50	200	JPEG	250
6	Huawei_Honor5c	C_6	4160 × 3120	90	160	JPEG	250
7	Samsung_GT-I8190 N	C_7	2560 × 1920	70	180	JPEG	250
8	Samsung_NV15_1	C_8	3648 × 2763	50	200	JPEG	250
9	Samsung_NV15_2	C_9	3648 × 2763	50	200	JPEG	250
10	Microsoft_Lumia640LTE	C_{10}	3264 × 1840	150	100	JPEG	250
11	Lenovo_P70A	C_{11}	7484 × 2704	150	100	JPEG	250

Fig. 7. An illustrative case for seam deletion and insertions.

Spyder (Python 3.8), and Ubuntu 64-bit.

A comparison of different versions of seam operations is shown in Fig. 7. In the resultant image the left-hand side corresponds to seam deletion, we can identify seam deletion in the area surrounded by red lines, which represents the gap between original and test images in a specific region. On the right is a description of the operation of adding seams. A red arrow indicates a difference between original and resultant pictures. There is a gap between certain regions. In both cases, manipulations have been accomplished without leaving any obvious traces behind.

Our analysis was conducted using two randomly selected sample images from the Camera: Microsoft_Lumia640LTE for seam deletion and Apple_iPadMini for seam insertion to determine the overall process and result. Here are the results for seam deletion and seam insertion in order to identify sources from which each seam was removed or inserted as shown in Tables 3 and 4.

Our example illustrates a complex problem and a solution through the seam function for insertion and deletion scenarios. We consider restricted forced seam carving to identify camera sources for Microsoft Lumia 640 LTE C_{10} in the seam deletion case and for Apple_iPad Mini C_1 in the seam addition case. As shown in Refs. [34,35], none of the uncarved blocks were smaller than 50 × 50 and 25 × 25. It appears that the method of selecting seams (e.g., randomly or based on energy) does not significantly affect anonymity. For our restricted force seams, we considered 20 × 20. In the next section, we will perform an ablation study to investigate the impact of the number of training images on the utilization of various groups. Additionally, we will conduct a correlation analysis using the Peak-to-Correlation Energy (PCE) metric.

4.5. Ablation analysis

In order to evaluate the performance of our proposed framework, an ablation study was conducted using Peak-to-Correlation Energy (PCE) [43] to investigate the effect of varying the number of groups used in the process of constructing camera fingerprints. Specifically, the experiment involved comparing the use of 10 groups (as in our current approach for detail see subsection 4.1) to the use of smaller or larger numbers of groups, such as 5 or 15 groups. The dataset was expanded from 10 groups (S_1–S_{10}) to 15 groups (S_1–S_{15}) for this purpose, with each group containing the same number of images. Additionally, in order to further assess the effect of group size on the results, the number of images within each group was also decreased from 250 to 130.

Table 3
(Seam deletion case) Feature values obtained from x_1, x_2 ….x_5 of different cameras when the test image is taken from Microsoft_Lumia640LTE.

S.#	Microsoft_Lumia640LTE	X1	X2	X3	X4	X5	Z
1	Apple_iPadMini	33.37145	1.055144	0	247.6577334	−5.33E-05	282.0842778
2	Canon_PowerShotA640_0	5.060317	0.861574	18	56.37233337	−0.00054	44.29368022
3	Sony_DSC-T77_1	−1.85209	−0.54172	123	−18.8541869	−3.54E-04	−144.248346
4	Xiaomi_RedmiNote3	−8.66611	−0.85745	156	−101.854791	−0.00056	−267.378904
5	Nikon_D200_0	28.52632	0.251814	22	217.6155562	−5.35E-05	224.3936317
6	Samsung_GT-I8190 N	11.6364	1.981307	76	129.9219562	−0.00029	67.53937228
7	D64_Samsung_NV15_0	3.950607	−0.00866	69	30.10350431	−7.59E-05	−34.9546207
8	D65_Samsung_NV15_1	6.654223	0.540678	76	64.16830448	−0.00012	−4.6369097
9	Huawei_Honor5c	5.458205	0.144227	48	55.40838742	−0.00018	13.01063906
10	**Microsoft_Lumia640LTE**	**49.1731**	**3.033618**	**0**	**395.4653928**	**5.35E-03**	**447.6774619**
11	Lenovo_P70A	0.922931	0.084157	76	6.497276331	−8.54E-05	−68.4957202

Table 4
(Seam Insertion case) Feature values obtained from x_1, x_2 ….x_5 of different cameras when the test image is taken from Apple_iPadMini camera.

S.#	Apple_iPadMini	X1	X2	X3	X4	X5	Z
1	**Apple_iPadMini**	**31.55307**	**4.309451659**	**0**	**297.3502**	**0.000568**	**333.2133254**
2	Canon_PowerShotA640_0	3.521448	0.222529594	108	37.00572	5.67E-06	−67.2502962
3	Sony_DSC-T77_1	21.13528	3.025148525	61	211.4655	−0.00018	174.6257931
4	Xiaomi_RedmiNote3	1.501889	0.489339933	134	19.11186	−6.79E-05	−112.896984
5	Nikon_D200_0	24.76095	1.709665244	35	239.9815	−0.00018	231.4519838
6	Samsung_GT-I8190 N	−6.32314	−2.541023804	22	−90.2946	−7.59E-05	−121.158823
7	D64_Samsung_NV15_0	−3.6521	−0.652708902	244	−37.4	−3.81E-05	−285.704858
8	D65_Samsung_NV15_1	−6.02129	−0.325261425	0	−91.3471	−0.73954	−98.4331871
9	Huawei_Honor5c	−7.48414	−3.514023804	66	−90.9264	−7.59E-05	−167.924623
10	Microsoft_Lumia640LTE	17.52113	−0.427156411	42	179.1918	−1.15E-04	154.2856518
11	Lenovo_P70A	−0.66213	0.654418494	54	−10.9457	−0.8652	−65.8186379

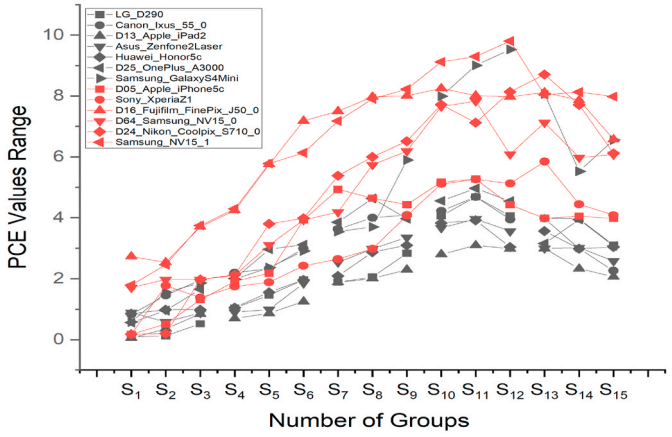


Fig. 8. Ablation study with block size of 30 × 30 and extended group size.

PCE is a widely-utilized method in the field of digital forensics for source camera identification due to its ability to effectively extract and compare image features. Given a correlation coefficient, the PCE between a fingerprint (F) and a Photo Response Non-Uniformity (PRNU) noise reference (R) can be calculated using following equation.

$$PCE = \frac{(peak(cc(K,R)))}{\sum (cc(K,R))^2} \tag{27}$$

where PCE is the Peak-to-Correlation Energy, K is the fingerprint,R is the PRNU noise,peak (cc (K,R)) is the peak value of the correlation coefficient between F and R, $\sum cc(K,R))^2$ is the total energy of the correlation coefficient between K and R and c (K,R) is the correlation coefficient between F and R.

We analyzed PCE values using a dataset of images which were divided into 30 × 30 blocks and grouped into 15 sets (labeled as S_1, S_2, … S_{15}) as shown in Fig. 8. The results, as shown in the figure above, indicated that the pattern of PCE values tended to increase until group S_{10}. However, from group S_{11} to S_{15}, the pattern became unstable and there was not a significant increase in PCE values. It is worth noting that the standard PCE value for successful identification ranges from 50 and above. This is why we combined the test image features in order to reach the standard acceptance. In addition to analyzing the full dataset, we also decreased the number of images in each group (each camera's image dataset was reduced to 130) in order to perform ablation studies. The purpose of these ablation studies was to assess the effect of this change on the results. Our analysis revealed that the ablation study had a strong correlation with the concepts presented in our approach. This information is useful for improving the accuracy and reliability of our framework in source attribution tasks.

In a further step, we propose a new approach that utilizes convolutional neural networks to learn features specific to each camera device as directly as possible from acquisition data. This data-driven algorithm is designed to improve the performance of camera-specific feature recognition.

5. One-dimensional convolutional neural network (1D-CNN)

To perform the experiments, we utilized the Adam optimizer with a learning rate of 10^{-3}. ReLU was selected as the activation function of the hidden layers, the last fully connected layer was activated using softmax with a loss function cross-entropy, and the numerical stability constant was $\varepsilon = 10^{-8}$. The mini-batch size was set to 16. As part of the training process, we randomly selected the training data to construct the mini-batch. In our proposed work, 35 epochs were trained, and we split the data into two parts: 80% for the training set and 20% for the validation set, which were used to locally evaluate our model performance and perform an ablation study. Our network was built using the Tensor Flow framework to construct, compile, and evaluate the model, and experiments were performed on Intel ® UHD Graphics 630.

5.1. Performance metrics

5.1.1. Seam deletion

A 1D-CNN model achieved 97% accuracy on a classification task using a softmax dense layer. Both training and validation with the loss function are shown in Fig. 9 for the seam deletion case.

5.1.2. Seam insertion

Next, we evaluated the dependence of the model quality from the training/validation and loss with respect to seam insertion. Additionally, we also observed that while the overall performance of the model on the seam insertion task was lower compared to the seam deletion task, the loss function values were similar between the two tasks as shown in following Fig. 10.

This suggests that the model may be struggling to learn the more complex patterns present in the seam insertion data, leading to lower accuracy. However, the similarity in the loss function values indicates that the model is still able to effectively learn and optimize the objective function in both tasks. Further analysis, such as examining the confusion matrix, may provide more insights into the model performance.

5.2. Overall performance analysis for both seam deletion and insertion

Our proposed IMG_{CAT} method was tested by seam-carrying 40 images from each camera with the restricted version (20 images for deletion and 20 images for insertion). For each of the 11 cameras, there are 440 images in total. The confusion matrices for the results obtained from

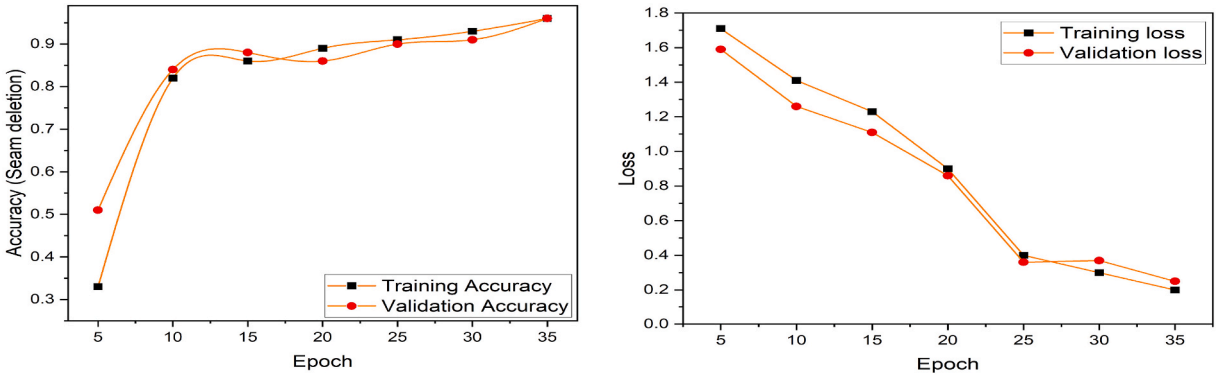


Fig. 9. Learning curves of the model loss and accuracy over each training epoch during training epoch and validation **a** Training accuracy vs. validation accuracy **b** Training loss vs. validation loss for seam insertion case.

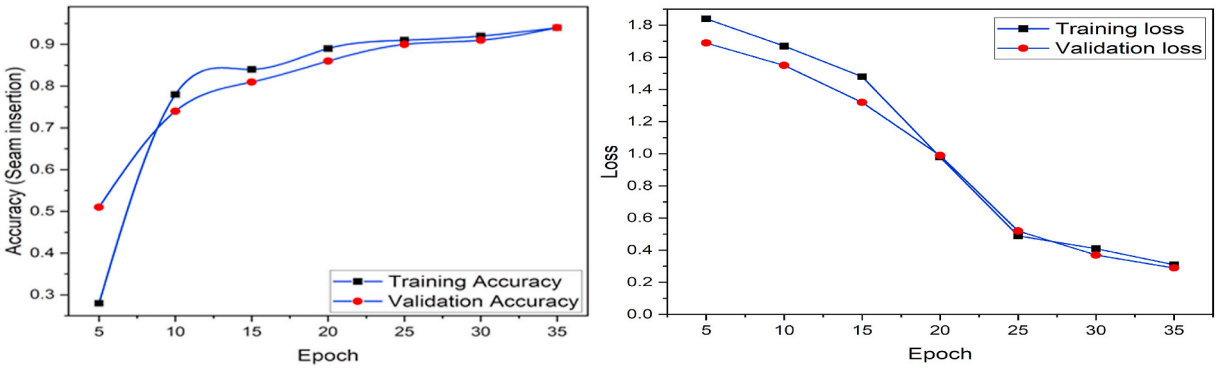


Fig. 10. Learning curves of the model loss and accuracy over each training epoch during training epoch and validation **a** Training accuracy vs. validation accuracy **b** Training loss vs. validation loss for seam deletion case.

Table 5
Confusion matrix for source identification of seam-carved (deletion) test images.

Labels	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C1	**89.56**	0.00	3.44	0.00	5.32	0.00	0.00	0.00	0.00	1.68	0.00
C2	0.00	**93.22**	0.00	0.00	3.11	0.26	0.00	0.00	0.00	0.00	3.41
C3	0.00	1.14	**96.01**	0.00	0.00	0.00	0.36	0.53	0.74	1.22	0.00
C4	0.00	0.00	0.56	**92.98**	0.45	2.58	0.00	0.00	0.00	0.00	3.43
C5	0.00	0.87	0.13	0.66	**95.29**	0.00	0.00	0.00	0.00	2.81	0.24
C6	0.00	0.25	0.41	2.11	0.00	**94.19**	0.00	0.89	0.53	0.00	1.62
C7	0.05	0.02	0.00	0.00	0.00	0.00	**99.17**	0.25	0.51	0.00	0.00
C8	0.03	0.16	0.00	0.00	0.07	0.00	1.10	**97.67**	0.97	0.00	0.00
C9	0.00	0.13	0.00	0.00	0.00	0.00	.41	1.36	**98.10**	0.00	0.00
C_{10}	2.22	0.15	0.00	0.00	0.63	0.00	0.00	0.00	0.00	**97.00**	0.00
C_{11}	0.00	0.00	0.00	1.02	0.00	0.84	0.00	0.00	0.00	0.00	**98.14**

Table 6
Confusion matrix for source identification of seam-carved (insertion) test images.

Labels	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
C1	**91.28**	0.00	2.36	0.00	6.07	0.00	0.00	0.00	0.00	0.29	0.00
C2	0.00	**90.89**	0.00	0.00	3.52	1.39	0.00	0.00	0.00	0.00	4.20
C3	0.00	0.46	**98.38**	0.00	0.00	0.00	0.09	0.27	0.04	0.76	0.00
C4	0.0	0.00	1.56	**94.05**	1.02	1.49	0.00	0.00	0.00	0.00	1.88
C5	0.00	1.51	0.39	0.07	**92.90**	0.00	0.00	0.00	0.00	4.02	1.11
C6	0.00	2.33	0.00	1.90	0.00	**91.19**	0.00	1.29	1.16	0.00	2.13
C7	1.27	0.58	0.00	0.00	0.00	0.00	**96.11**	1.05	0.99	0.00	0.00
C8	0.21	0.77	0.00	0.00	0.36	0.00	2.26	**94.79**	1.61	0.00	0.00
C9	0.00	0.46	0.00	0.00	0.00	0.00	1.19	1.70	**96.65**	0.00	0.00
C10	1.61	0.01	0.00	0.00	1.28	0.00	0.00	0.00	0.00	**97.10**	0.00
C11	0.00	0.00	0.00	1.71	0.00	2.98	0.00	0.00	0.00	0.00	**95.31**

the proposed IMG_{CAT} for phase I are shown in Tables 5 and 6. To show the performance of the algorithm, we display the actual and predicted instances of occurrences. The matrix columns represent the predicted classes, while the rows represent the actual classes. The confusion matrix provides an easy method to analyze data, since all true positives will lie on the primary diagonal of the confusion matrix.

Confusion **matrices** are presented in Tables 5 and 6 for each of the eleven cameras in the identification of random test images from the dataset. We tested images from various backgrounds (light, dark, object orientations, indoor, outdoor, etc.). The average accuracy was calculated for each camera device based on 20 test images. For seam deletion, the average accuracy was 89–99%. For seam insertion, 90% and 98% were recorded. Some cameras have relatively low accuracy because they are subjected to unfavorable conditions such as dark backgrounds and blurry pictures. Median accuracies of 95.58% and 94.42% were achieved. All metrices are derived using following formulas.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{28}$$

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{29}$$

$$\text{Cross} - \text{entropy loss} = -\sum(y \times \log(\hat{y}) + (1 - y) \times \log(1 - \hat{y})) \tag{30}$$

where TP (true positive) represents the number of correct positive predictions made by the model or system, TN (true negative) represents the number of correct negative predictions made by the model or system, FP (false positive) represents the number of incorrect positive predictions made by the model or system, and FN (false negative) represents the number of incorrect negative predictions made by the model or system. Similarly y is true label and $\hat{y}$ is the predicted probability of the true label.

5.3. Discuss and comparison analysis

Convolutional neural network (CNN) have been widely used for seam deletion and insertion tasks due to their ability to learn complex patterns in visual data. However, most of the existing literature on the

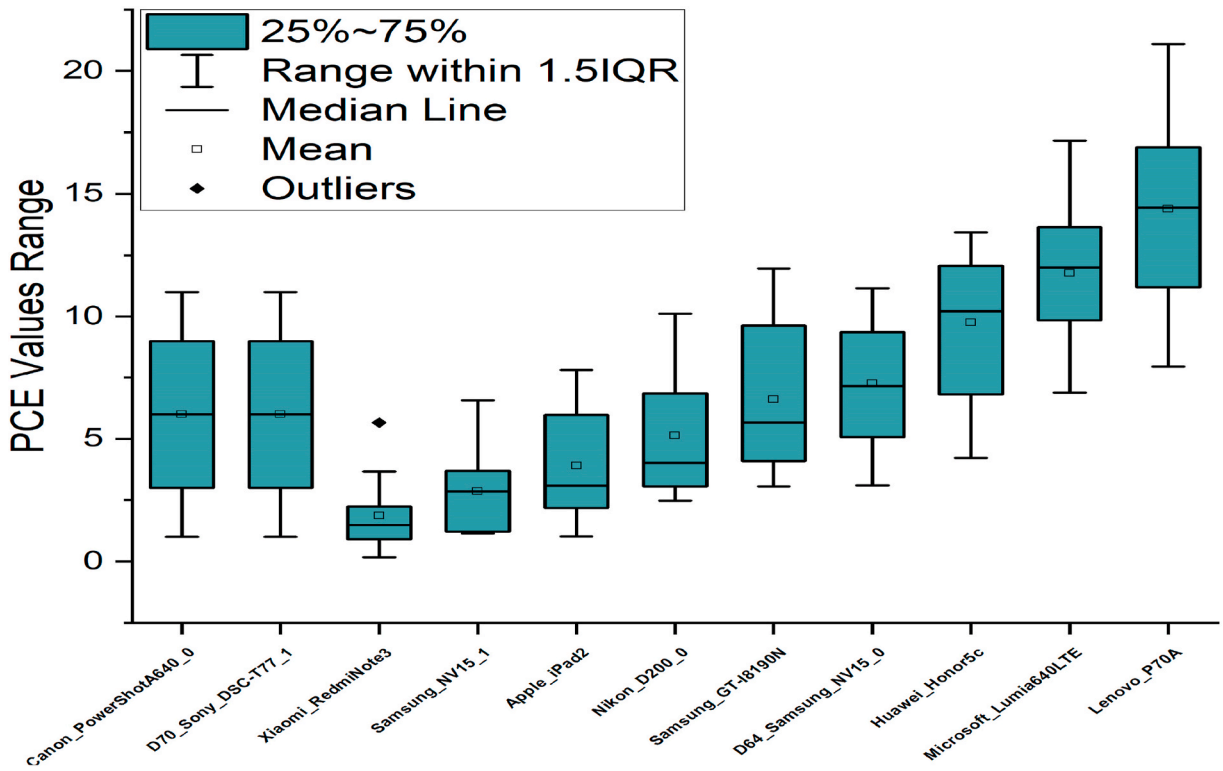


Fig. 11. A comparison of our study results.

use of CNN for these tasks has focused on seam deletion [8,44–46], with relatively few studies investigating the use of CNN for seam insertion [8]. This is likely due to the greater complexity of the seam insertion task, which requires the model to synthesize new pixels rather than simply removing or modifying existing ones. To the best of our knowledge, there have been no studies in the literature that have specifically examined the use of CNN for source camera identification in the context of seam deletion and insertion. Most of the existing research in this area has focused on using CNNs for seam detection, with relatively little attention given to the use of CNNs for source camera identification. In light of the fact that a significant portion of previous literature has employed PCE as a metric for source camera attribution [10,47–50]. We deemed it appropriate to conduct an examination of our proposed approach using this metric on 50 × 50 blocks. The results of this evaluation revealed the efficacy of this approach.

In Fig. 11 box plot shows the distribution and density of the data, while the box plot shows the summary statistics and potential outliers. This combination can be useful for comparing multiple data sets and identifying patterns and trends in the data. In this case, the PCE values range from 0 to 25. The first quartile (Q1) is the value that separates the lowest 25% of the data from the highest 75%. The median is represented by a line inside the box. Outliers, which are data points that are significantly different from the rest of the data. Based on the results of our study, it can be inferred that source camera attribution is not viable when the provided image has undergone seam-based manipulation. As per previous literature, the standard value of PCE for successful camera attribution is above 50. However, as a result of the utilization of seam operations, the obtained PCE value was substantially lower at approximately 25. This finding serves as a justification for the utilization of a combination of feature extraction methods in evaluating the overall performance, as it accounts for the limitations in utilizing PCE as the sole metric in scenarios involving manipulated images. Ultimately, it is important to acknowledge the limitations of this study when interpreting the results. Notably, certain camera models may possess patterns that are not consistent with the uptrending condition observed in this investigation. As such, further research is required to assess and address these limitations, in order to attain a more comprehensive understanding of the potential and limitations of source camera identification.

6. Conclusions

A challenging problem in seam-carved resized images is identifying the source camera. Current work identifies the block by finding 1) an uncarved 50 × 50 block in a seam-carved image or 2) multiple uncarved blocks smaller than 50 × 50 from one or several seam-carved images (only deletion case). In this paper, we examined the reliability of source camera identification when considering the deletion and insertion aspects of a query image. It is achieved if an uncarved block is smaller than 50 × 50 in a single image. We tested our proposed method IMG_{CAT} on a dataset containing smartphones and digital cameras. Using camera fingerprints to characterize a device may be more reliable than a single PRNU, as we found in an observation. Then, we extracted five features from the correlation values with the series of PRNUs that enabled us to link seam images (deletion and insertion) to their source device. In the next phase, we fused the 1D CNN to validate IMG_{CAT} based on the results recorded from feature extractions. The results prove that the proposed method worked even when the block size was 20 × 20 pixels. As part of future work, two parallel lines of research will be followed. Initially, we will explore how we can specialize the CNN to learn additional characteristics as well as traces of camera models (e.g. scaling, rotation, blurring, etc.). Meanwhile, how to utilize LSTM capabilities directly for localization by training with split layers and corresponding mitigation strategies on synthetic images.

CRediT author statement

LAW Ngai-Fong, Bonnie: Conceptualization, Methodology, Software, **Muhammad Irshad**: Data curation, Writing- Original draft preparation. **Ka Hong Loo**: Visualization, Investigation. **LAW Ngai-Fong, Bonnie**: Supervision. **Muhammad Irshad**: Software,

Validation. **Sami Haider**: Formal analysis. **LAW Ngai-Fong, Bonnie**: Reviewing and Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the GRF Grant 15211720, (project code: Q79 N), of the Hong Kong SAR Government. Muhammad Irshad Ibrahim would like to thank the postdoctoral fellowship support from the Hong Kong Polytechnic University (GYW4X).

References

[1] Dimitriadis A, Ivezic N, Kulvatunyou B, Mavridis I. D4I - digital forensics framework for reviewing and investigating cyber attacks. Array 2020;5:100015. https://doi.org/10.1016/J.ARRAY.2019.100015.
[2] Liu Y, Zou Z, Yang Y, Law NFB, Bharath AA. Efficient source camera identification with diversity-enhanced patch selection and deep residual prediction. Sensors 2021;21(14):4701. https://doi.org/10.3390/S21144701. *Page 4701*.
[3] Irshad M, et al. City vision: CCTV images based public surveillance model. In: Proc. - 2021 Int. Conf. Electron. Inf. Technol. Smart Agric. ICEITSA; 2021. p. 416–20. https://doi.org/10.1109/ICEITSA54226.2021.00085.
[4] Law SC, Law NF. PRNU-based source identification for network video surveillance system. In: 2018 IEEE Int. Conf. Consum. Electron. ICCE; 2018. p. 1–2. https://doi.org/10.1109/ICCE.2018.8326094. 2018-January Mar. 2018.
[5] Nowroozi E, Dehghantanha A, Parizi RM, Choo KKR. A survey of machine learning techniques in adversarial image forensics. Comput Secur 2021;100:102092. https://doi.org/10.1016/J.COSE.2020.102092.
[6] Bhima Koregaon case: Forensics report states evidence was planted in case against Indian activists accused of plotting to overthrow the Modi government - the Washington Post." https://www.washingtonpost.com/world/asia_pacific/india-bhima-koregaon-activists-jailed/2021/02/10/8087f172-61e0-11eb-a177-7765f29a9524_story.html (accessed Jan. 27, 2022).
[7] Avidan S, Shamir A. Seam carving for content-aware image resizing. Proc ACM SIGGRAPH Conf Comput Graph 2007;26. https://doi.org/10.1145/1275808.1276390.
[8] Nam SH, Ahn W, Yu IJ, Kwon MJ, Son M, Lee HK. Deep convolutional neural network for identifying seam-carving forgery. IEEE Trans Circ Syst Video Technol 2021;31(8):3308–26. https://doi.org/10.1109/TCSVT.2020.3037662.
[9] Dirik AE, Sencar HT, Memon N. Analysis of seam-carving-based anonymization of images against PRNU noise pattern-based source attribution. IEEE Trans Inf Forensics Secur 2014;9(12):2277–90. https://doi.org/10.1109/TIFS.2014.2361200.
[10] Shi C, Law NF, Leung FHF, Siu WC. A local variance based approach to alleviate the scene content interference for source camera identification. Digit Invest Sep. 2017;22:74–87. https://doi.org/10.1016/J.DIIN.2017.07.005.
[11] Chai D. SQL: superpixels via quaternary labeling. Pattern Recogn 2019;92:52–63. https://doi.org/10.1016/J.PATCOG.2019.03.012.
[12] RubinsteinMichael ShamirAriel, AvidanShai. Improved seam carving for video retargeting. ACM Trans Graph 2008;27(3). https://doi.org/10.1145/1360612.1360615.
[13] Shamir A, Avidan S. Seam carving for media retargeting. Commun ACM 2009;52(1):77–85. https://doi.org/10.1145/1435417.1435437.
[14] Frankovich M, Wong A. Enhanced seam carving via integration of energy gradient functionals. IEEE Signal Process Lett 2011;18(6):375–8. https://doi.org/10.1109/LSP.2011.2140396.
[15] Popescu AC, Farid H. Exposing digital forgeries by detecting traces of resampling. IEEE Trans Signal Process 2005;53(2 II):758–67. https://doi.org/10.1109/TSP.2004.839932.
[16] Han R, Ke Y, Du L, Qin F, Guo J. Exploring the location of object deleted by seam-carving. Expert Syst Appl 2018;95:162–71. https://doi.org/10.1016/J.ESWA.2017.11.023.
[17] Ryu SJ, Lee HY, Lee HK. Detecting trace of seam carving for forensic analysis. IEICE Trans Info Syst 2014;E96(5):1304–11. https://doi.org/10.1587/TRANSINF.E97.D.1304.
[18] Rashid A, Peng Y, Rooh UA, Muhammad I. Image denoising using wavelet transform. Front Artif Intell Appl 2019;314:142–9. https://doi.org/10.3233/978-1-61499-939-3-142.
[19] Zhou T, Ruan S, Canu S. A review: deep learning for medical image segmentation using multi-modality fusion. Array 2019;3(4):100004. https://doi.org/10.1016/J.ARRAY.2019.100004.
[20] Yoon M, Nam SH, Yu IJ, Ahn W, Kwon MJ, Lee HK. Frame-rate up-conversion detection based on convolutional neural network for learning spatiotemporal features. Forensic Sci Int 2022;340:111442. https://doi.org/10.1016/J.FORSCIINT.2022.111442.
[21] Rai Y, Le Callet P. Visual attention, visual salience, and perceived interest in multimedia applications. Acad Press Libr Signal Process Image Video Process Anal Comput Vis 2018;6:113–61. https://doi.org/10.1016/B978-0-12-811889-4.00003-8.
[22] Sarkar A, Nataraj L, Manjunath BS. Detection of seam carving and localization of seam insertions in digital images. In: MMandSec'09 - Proc. 11th ACM Multimed. Secur. Work.; 2009. p. 107–16. https://doi.org/10.1145/1597817.1597837.
[23] Da Wei J, Lin YJ, Wu YJ. A patch analysis method to detect seam carved images. Pattern Recogn Lett 2014;36(1):100–6. https://doi.org/10.1016/J.PATREC.2013.09.026.
[24] Chan LH, Law NF, Siu WC. A two dimensional camera identification method based on image sensor noise. ICASSP, IEEE Int Conf Acoust Speech Signal Process - Proc 2012:1741–4. https://doi.org/10.1109/ICASSP.2012.6288235.
[25] Boroumand M, Chen M, Fridrich J. Deep residual network for steganalysis of digital images. IEEE Trans Inf Forensics Secur 2019;14(5):1181–93. https://doi.org/10.1109/TIFS.2018.2871749.
[26] Zhang WN, Liu YX, Zhou J, Yang Y, Law NF. An improved sensor pattern noise estimation method based on edge guided weighted averaging. Lect Notes Comput Sci 2020;12487:405–15. https://doi.org/10.1007/978-3-030-62460-6_36.
[27] Garg A, Singh AK. Analysis of seam carving technique: limitations, improvements and possible solutions. Vis. Comput.; Apr. 2022. p. 1–27. https://doi.org/10.1007/S00371-022-02486-2/FIGURES/24.
[28] Bayar B, Stamm MC. Constrained convolutional neural networks: a new approach towards general purpose image manipulation detection. IEEE Trans Inf Forensics Secur 2018;13(11):2691–2706, Nov. https://doi.org/10.1109/TIFS.2018.2825953.
[29] Nam SH, Park J, Kim D, Yu IJ, Kim TY, Lee HK. Two-stream network for detecting double compression of H.264 videos. Proc - Int Conf Image Process ICIP Sep. 2019:111–5. https://doi.org/10.1109/ICIP.2019.8802966.
[30] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.; 2015.
[31] Russakovsky O, et al. ImageNet large scale visual recognition challenge. Int J Comput Vis Dec. 2015;115(3):211–52. https://doi.org/10.1007/S11263-015-0816-Y/FIGURES/16.
[32] Ye J, Shi Y, Xu G, Shi YQ. A convolutional neural network based seam carving detection scheme for uncompressed digital images. Lect Notes Comput Sci 2019;11378:3–13. https://doi.org/10.1007/978-3-030-11389-6_1.
[33] Lukáš J, Fridrich J, Goljan M. Digital camera identification from sensor pattern noise. IEEE Trans Inf Forensics Secur Jun. 2006;1(2):205–14. https://doi.org/10.1109/TIFS.2006.873602.
[34] Bayram S, Sencar HT, Memon ND. Seam-carving based anonymization against image & video source attribution. In: 2013 IEEE Int. Work. Multimed. Signal Process. MMSP; 2013. p. 272–7. https://doi.org/10.1109/MMSP.2013.6659300.
[35] Taspinar S, Mohanty M, Memon N. PRNU-based camera attribution from multiple seam-carved images. IEEE Trans Inf Forensics Secur 2017;12(12):3065–3080, Dec. https://doi.org/10.1109/TIFS.2017.2737961.
[36] Li W, Xie Y, Zhou H, Han Y, Zhan K. Structure-aware image fusion. Optik 2018;172:1–11. https://doi.org/10.1016/J.IJLEO.2018.06.123.
[37] Ssengonzi C, Kogeda OP, Olwal TO. A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization. Array 2022;14:100142. https://doi.org/10.1016/J.ARRAY.2022.100142.
[38] Irshad M, Law NF, Loo KH. CamCarv - Expose the Source Camera at the Rear of Seam Insertion. In: Rutkowski L, Scherer R, Korytkowski M, Pedrycz W, Tadeusiewicz R, Zurada JM, editors. Artificial Intelligence and Soft Computing. Lecture Notes in Computer Science, 13589. Cham: Springer; 2023. https://doi.org/10.1007/978-3-031-23480-4_2.
[39] Baldini G, Steri G. A survey of techniques for the identification of mobile phones using the physical fingerprints of the built-in components. IEEE Commun. Surv. Tutorials 2017;19(3):1761–1789, Jul. https://doi.org/10.1109/COMST.2017.2694487.
[40] Schmidhuber J. Deep learning in neural networks: an overview. Neural Network Jan. 2015;61:85–117. https://doi.org/10.1016/J.NEUNET.2014.09.003.
[41] Gloe T, Böhme R. The 'dresden image database' for benchmarking digital image forensics. In: Proc. 2010 ACM Symp. Appl. Comput. - SAC '10; 2010. https://doi.org/10.1145/1774088.
[42] Shullani D, Fontani M, Iuliani M, Al Shaya O, Piva A. VISION: a video and image dataset for source identification. EURASIP J Inf Secur Oct. 2017;2017(1):1–16. https://doi.org/10.1186/S13635-017-0067-2.
[43] Ma H, Acton ST, Lin Z. SITUP: scale invariant tracking using average peak-to-correlation energy. IEEE Trans Image Process 2020;29:3546–57. https://doi.org/10.1109/TIP.2019.2962694.
[44] Iqbal M, Chen L, Fu H, Lin Y. Seam carve detection using convolutional neural networks. Lect Notes Inst Comput Sci Soc Telecommun Eng LNICST 2019;279:392–407. https://doi.org/10.1007/978-3-030-19086-6_44/COVER.
[45] Cieslak LFS, Da Costa KA, Paulopapa J. Seam carving detection using convolutional neural networks. In: SACI 2018 - IEEE 12th Int. Symp. Appl. Comput. Intell. Informatics, Proc.; Aug. 2018. p. 195–9. https://doi.org/10.1109/SACI.2018.8441016.

[46] Ahmadi M, Karimi N, Samavi S. Context-aware saliency detection for image retargeting using convolutional neural networks. Multimed Tool Appl 2021;80(8):11917–41. https://doi.org/10.1007/S11042-020-10185-0/METRICS.
[47] Karaküçük A, Dirik AE, Sencar HT, Memon ND. Recent advances in counter PRNU based source attribution and beyond. Media Watermark Secur Foren 2015;9409:94090N. https://doi.org/10.1117/12.2182458.
[48] Costa F De O, Silva E, Eckmann M, Scheirer WJ, Rocha A. Open set source camera attribution and device linking. Pattern Recogn Lett Apr. 2014;39(1):92–101. https://doi.org/10.1016/J.PATREC.2013.09.006.
[49] Zhao Y, Zheng N, Qiao T, Xu M. Source camera identification via low dimensional PRNU features. Multimed Tool Appl 2018;787(7):8247–69. https://doi.org/10.1007/S11042-018-6809-4.
[50] Jahanirad M, Wahab AWA, Anuar NB. An evolution of image source camera attribution approaches. Forensic Sci Int 2016;262:242–75. https://doi.org/10.1016/J.FORSCIINT.2016.03.035.