Signal Processing: Image Communication 68 (2018) 162–168

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Source camera model identification based on convolutional neural networks with local binary patterns coding

Bo Wang [a,*], Jianfeng Yin [a], Shunquan Tan [b], Yabin Li [a], Ming Li [a]

[a] School of Information and Communication Engineering, Dalian University of Technology, China
[b] College of Computer Science and Software Engineering, Shenzhen University, China

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Camera model identification
Image forensics
Convolutional neural networks (CNNs)
Local binary patterns (LBP)

ABSTRACT

Source camera model identification has always been one of the main fields of digital image forensics since it is the foundation of solving a wide range of forensic problems. Several effective camera model identification algorithms have been developed for the practical necessity. However, they are mostly based on traditional machine learning methods and rely on well-designed features or models. Since deep learning has made great progress in computer vision tasks, significant interest has arisen in applying deep learning in image forensics. In this paper, we present a deep learning approach to tackle source camera model identification problem. We modify a convolutional neural networks (CNNs) structure similar to AlexNet and equip it with a simple local binary patterns (LBP) preprocessing operation. The identification accuracy on the public database "Dresden Image Database" achieves 98.78% over 12 camera models without any other sophisticated procedures, for instance, extra classifier, majority voting, etc.

1. Introduction

With the development of information technology and social media networks, digital images are found everywhere in our daily life and nearly become the most pervasive information carrier. Besides playing an important role in information spreading, digital images, as a kind of visual data, are usually regarded as a certification of truth or evidences in front of a court of law since we traditionally believe in the integrity of what we see. However, people can acquire images easily with the popularity of inexpensive cameras and cell phone devices, and more importantly, can also manipulate them from the source information to contents and even create ones as they want with the development of image manipulation softwares and social networks over the years. Therefore, the situation highlights the necessity to verify the source and authenticity of digital images. It is a key work in the field of digital image forensics [1].

As one main field of image forensics, source camera identification has two branches. One is to match an image with one individual camera, and the other is to match it with a specific camera model. For these tasks, the researchers have been devoted to studying the pipeline of image acquisition process and exploiting traces or artifacts introduced to the images to capture source information. As shown in Fig. 1, image acquisition process involves several stages each of which can be implemented differently in different cameras. Consequently, some unique traces or artifacts are introduced in the final images. According to those traces from given cameras, a variety of camera identification approaches have been proposed and they can be categorized into two groups.

The first group of methods managed to compute a hypothesized analytical model on certain stages of image acquisition and then evaluates the correlation between the model and the tested image. Lucas et al. [2] used a sensor pattern noise model to identify the source camera sensors. Choi et al. [3] chose a lens radial distortion pattern as a fingerprint. Dirik et al. [4] utilized dust-spot characteristics to identify the source digital single lens reflex camera. In order to extract reliable photo-response non-uniformity (PRNU), Amerini et al. [5] introduced a minimum mean square error (MMSE) filter in the un-decimated wavelet domain to estimate the PRNU noise. Li [6] suggested that an enhanced fingerprint can be obtained by assigning weighting factors according to the magnitude of scene details to eliminate the influence of image content. Furthermore, Tomioka et al. [7] proposed a method based on the pairwise relationships of pixel clusters to suppress the effects of noise contamination. Recently, Li et al. [8] proposed the use of principal component analysis (PCA) to formulate a compact SPN representation. Besides, a training set construction procedure was also proposed to enhance the de-noising effect in [8].

* Corresponding author.
E-mail addresses: bowang@dlut.edu.cn (B. Wang), jfyin@pku.edu.cn (J. Yin), tansq@szu.edu.cn (S. Tan), yabinli@mail.dlut.edu.cn (Y. Li), mli@dlut.edu.cn (M. Li).

https://doi.org/10.1016/j.image.2018.08.001
Received 2 December 2017; Received in revised form 21 June 2018; Accepted 2 August 2018
Available online 4 August 2018

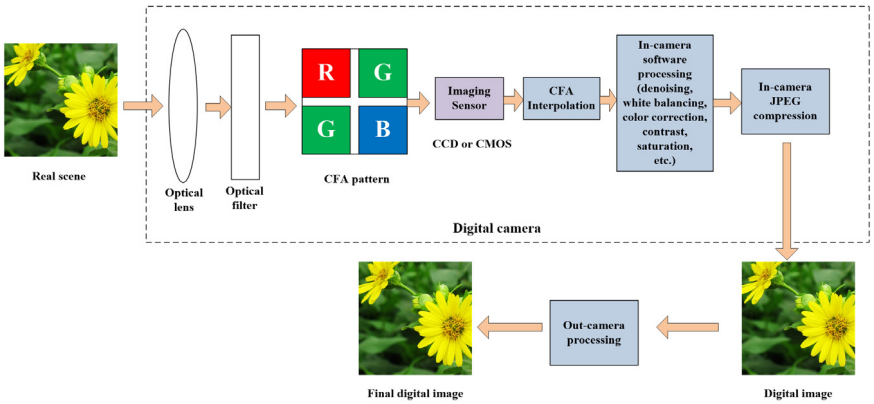

Fig. 1. A common digital image acquisition process. When a digital camera captures an image, the light reflected by the real scene firstly pass through the optical lens and the filter. Before hitting the imaging sensor which is usually only one and can only record on color value at each pixel, a color filter array(CFA) is used to allow only one color component of light to pass through it at each position. Then the other two color channels are interpolated by specific CFA interpolation algorithm which is known as demosaicing algorithm. After demosaicing, a set of post-processing operations including software processing and in-camera JPEG compression are performed to obtain a digital image. Finally, the image experiences some out-camera processing such as data transmission and computer processing to generate the final digital image.

The other group of methods relied on well-designed feature vector extraction and machine learning classifiers. Swaminathan et al. [9] constructed an efficient camera identifier through estimating interpolation coefficients of color filter array (CFA). Xu et al. [10] extracted 354-dimensional features based on local binary patterns (LBP) to distinguish camera models. Hu et al. [11] developed an improved algorithm using inter-channel demosaicing traces for camera model identification. Tuama et al. [12] proposed a method to extract high order statistics consisted of co-occurrences matrix, traces of color dependencies features related to CFA interpolation arrangement, and conditional probability statistics. A better identification result compared with the correlation based method was reported in their paper. Different from [10], the recent work [13] also investigated the discriminative ability of local phase quantization (LPQ), a LBP like texture descriptor, to distinguish imaging devices. The combined texture features of LBP and LPQ resulted in higher identification accuracy compared with [10]. Chen *et al* [14] built a rich model of 1372-dimensional features to identify the model of an image's source camera. They utilized two co-occurrence matrixes to capture the reconstructed error between the original image and the reconstructed version. The average identification accuracy of 99.2% over 12 camera models was reported in their paper.

On the other hand, the convolutional neural networks (CNNs) ,which are strong in feature learning, have made great success in computer vision tasks and developed rapidly since 2012 [15–18]. These achievements arouse attention from the community of digital image forensics and several works have been done to exploit suitable approaches to apply CNNs to solve forensic problems. Baroffio et al. firstly applied CNNs to identify source cameras, but the poor performance indicated that the CNNs designed for computer vision (CV) cannot be suited to camera identification directly. On the basis, they proposed a new approach to dealing with camera identification in [19]. They simply regarded CNNs as a feature extractor and combined the networks with SVM classifiers to complete the classification tasks. Chen et al. [20] noticed that the difference among classes of image forensics problems is subtle and added a preprocessing layer before CNNs for median filtering forensics. This preprocessing process achieved a significant boost in performance. Bayer et al. [21] proposed a new convolutional layer which is similar to a preprocessing layer as a part of the CNNs to detect image manipulation. Tuama et al. [22] presented a CNNs structure similar to AlexNet and equipped it with a high-pass filter (HPF) layer [23] to cope with camera model identification. Their experimental results showed the important role that preprocessing part plays in classification accuracy and indicated that trying the bigger networks such as GoogleNet or ResNet might be promising. The experimental results in [24] also confirmed the points mentioned above. Recently, Yang et al. [25] proposed content-adaptive fusion residual networks to detect image origin and achieved satisfactory performances in the case of query images with small size in camera brand identification. But for camera model identification, the detection accuracy is only 87.55%, so there is still much room for improvement.

The CNNs method is promising but not efficient enough for camera model identification. Firstly, the database available is not as large as ImageNet. Therefore, the CNNs cannot be very deep, or rather the effectiveness of training may be affected. Secondly, the CNNs tailored for CV are sensitive to the primary visual information of an image rather than the intrinsic source information which is invisible for the naked eyes. From the view of the current situation of the related research, the intrinsic source information is mostly hidden in the noise of images or other statistical features. It might be not realistic to let CNNs capture such low signal-to-noise signals. However, we can make "glasses" for CNNs so that the effective source information can be enlarged in the "eye" of CNNs or exists in a way that is easier to be recognized by CNNs. That is to say, we can give some hints or guidance to CNNs. Therefore, we propose that CNNs can capture the information that human's eyes cannot perceive and accomplish the camera model identification tasks with the help of research results from the experts of related fields. Particularly, we modify a shallow CNNs architecture like AlextNet and use the simplest LBP operator to help the CNNs to extract source camera information of images. Experimental studies illustrate that our method achieves better classification accuracy results compared with the state-of-art classical and CNN-based methods.

The rest of the paper is organized as follows. Section 2 explains the details of our proposed method. In Section 3, extensive experiments are carried out and comparisons of state-of-art are presented to show the superiority of proposed method. And, conclusions of the paper are drawn and some perspectives are proposed in Section 4.

2. Proposed method

In this section, we firstly present the description of coding preprocessing operation which is applied before the CNNs architecture based on local binary patterns. Then the CNNs architecture as well as the design considerations are introduced in details . Finally, the implementation details of training and testing process are presented including our selection or adjustment strategies of some key parameters. The framework of the proposed method is illustrated in Fig. 2.

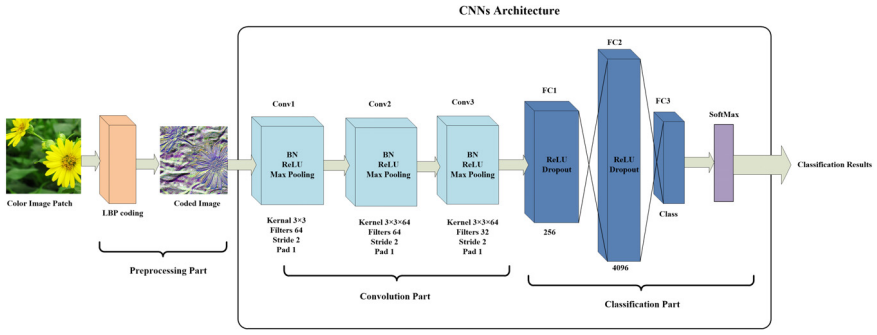

Fig. 2. The layout of the proposed method. The main part of the structure is the CNNs architecture similar to AlexNet. And it is equipped with a simple LBP operation preprocessing in front of the first convolutional layer. The CNNs architecture is composed of 3 convolutional layers and 3 fully connected layers. A color image is coded to LBP map by the LBP coding operation firstly and then fed into CNNs. Convolutional layers calculate the convolutions of the LBP maps and do Batch Normalization operation of the results before they are fed into activation function ReLU. Finally, the activation results are processed by a max pooling layer and go to the next layer. According to the feature maps extracted by the convolutional layers, the fully connected layers can complete the classification task.

2.1. LBP coding operation

Local binary pattern, which was firstly proposed by Ojala [26] in 1996, is a simple but efficient texture operator at the beginning. It labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. According to [27], the LBP value of pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{1}$$

where

$$s(x) = \begin{cases} 1, if\ x \geq 0 \\ 0, otherwise \end{cases} \tag{2}$$

where R denotes the radius of a circularly symmetric neighborhood used for LBP calculation, and P denotes the number of the samples around the circle. The gray levels of the center pixel and its neighbor pixels are represented as g_c and g_p, respectively.

In [10], an approach for camera model identification using local binary patterns is proposed and achieved excellent performance. However, the 354-dimensional features are selected from 1536-dimensional LBP features of three different transformation domains and two color channels of the given image, which is a sophisticated and time-consuming procedure. Considering that CNNs is a good feature extractor and classifier, the LBP map of an image is likely to be a good hint for CNNs because the features based on LBP are efficient for camera model identification. It is noteworthy that LBP operation suppresses the interference caused by the contents of the image and forces the CNNs to focus more on intrinsic characteristics of the query image rather than the contents or other primary vision information. Therefore, a LBP operator, whose parameters are acquiescently set as $R=1$ and $P=8$, is adopted to compute the LBP maps of a color image according to Eqs. (1) and (2), inspired by the algorithm in [10]. It is worth noting that the LBP maps of an image are computed in three color channels independently and then are fed into CNNs as inputs. An example for the LBP value computing process of a pixel from one channel is illustrated in Fig. 3.

2.2. The CNNs architecture

In order to satisfy the demand of camera model identification, we modify a CNNs architecture which resembles to AlexNet. Fig. 4 reports the characteristics and parameters of the CNNs architecture that we use.

2.2.1. Convolutional layers

Generally, a convolutional layer contains four sub-layers: convolution, activation function, pooling and normalization. The convolution can be calculated by the following formula:

$$a_j^l = \sum_{i=1}^{n} a_i^{l-1} * w_{i,j}^{l-1} + b_j^l \tag{3}$$

where $*$ denotes convolution and n is the number of feature maps of layer l-1. a_j^l denotes the jth output map of layer l and it is used to execute convolution with the convolution kernel. $w_{i,j}^{l-1}$, as the convolution kernel, connects the ith output of layer l-1 and the jth input of layer l. As for b_j^l, it is the bias parameter for the jth output map of layer l.

Then the activation function acts on each value of the last layer's output map. Among several kinds of activation functions (e.g. $tanh(x)$, $sigmoid(x)$, etc.), Rectified Linear Units (ReLU) function [28] is the most popular one and it can be represented as follows:

$$f(x) = max(0, x) \tag{4}$$

where x is the activation value of ReLU.

According to the results in [15], the CNNs equipped with the non-saturating nonlinearity ReLU can be trained much faster than those with *tanh* units or other saturating nonlinearities.

Pooling [29] layers in CNNs are usually comprised to summarize the joint distribution of neighboring groups of neurons, reduce the resolution of feature maps and even slightly reduce over-fitting. Experiments in [15,22] have proved that the influence of pooling layers after the output maps of neurons is significant. Hence, we apply overlap max pooling layers after activation functions of convolution layers.

Local response normalization (LRN) [15] is commonly applied after the ReLU nonlinearity in certain convolutional layers of AlexNet. However, we introduce batch normalization (BN) [30] in our networks. It is not only because BN is more promising in the latest artificial intelligence tasks but also we have some considerations which will be expressed particularly in the next sub-section.

2.2.2. Batch normalization

Internal covariate shift (ICS) [30], which refers to that the distributions of the inputs of hidden layers are always changing along with the variation of network parameters during the training process, has a negative impact on model training process and convergence. Batch normalization inspired by whitening activations is an efficient technique to address ICS issue and beneficial to CNN training. The core idea of batch normalization is to ensure that the network always produce activations with desired distribution which is Gaussian distribution in general. Let x be an input of a layer, treated as a feature map, and χ be the set of these input over the training data set. The normalization can be written as a transformation

$$\hat{\mathrm{x}} = Norm(\mathrm{x}, \chi) \tag{5}$$

As for a mini-batch input B of size m: $B = (x_1, x_2, \ldots x_m)$, the normalization of each dimension is:

$$\hat{x}_i = \frac{x_i - E[x_i]}{\sqrt{Var[x_i]}} \tag{6}$$

where the expectation and variance are computed over a mini-batch of the training data set.

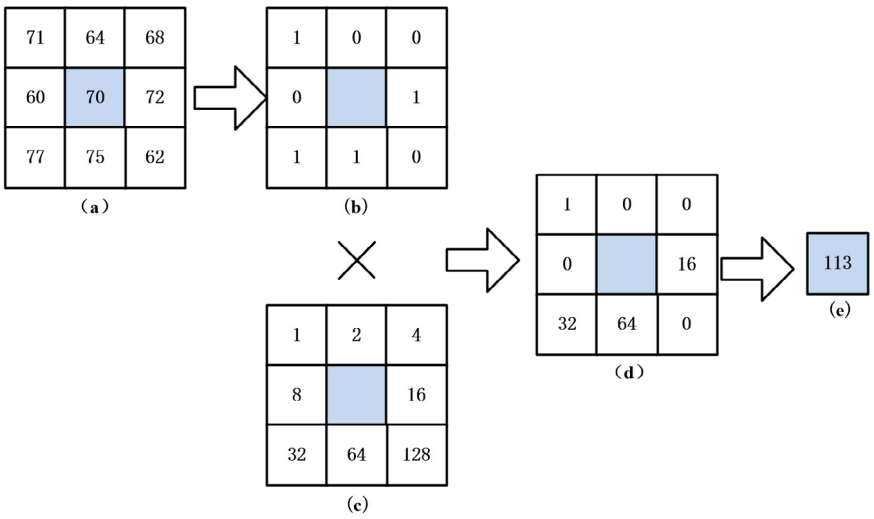

Fig. 3. An example for the LBP value computing process of a pixel. (a)The pixel values of an image neighborhoods. (b) The result after thresholding the neighborhood of each pixel. (c) The weight of each position. (d) Consider the results as binary numbers. (e) The LBP value result of the pixel in the middle.

Then a pair of parameters γ, β are introduced for each activation x_i to scale and shift the normalized value:

$$y_i=\gamma\hat{x}_i+\beta \tag{7}$$

The parameters γ, β can be learned and adjusted as other parameters during the training process.

Through batch normalization, the input distribution of each hidden layers' neuron which may move to saturated region of some nonlinearities gradually is forced to come back to standard Gaussian distribution so that the activations can fall into the sensitive region of the nonlinearity and then the gradient vanishing problem can be avoided as a result [30].

In our model, batch normalization is performed after each convolution for the following three reasons:

(i) BN prevents the training process from falling into local minima and over-fitting, and helps the networks find optimal scales and biases for feature maps [31].

(ii) BN acts as a regularizer and accelerates the training process which refers to higher convergence speed and lower final error rate.

(iii) BN makes the networks more tolerant to the network parameters and yield in models with better generalization ability, which means we can be less careful about initialization.

2.2.3. Classification layers

The classification layers are composed of three fully connected layers and a softmax function which is replaced by softmaxloss function in training stage. Like AlexNet, the first two fully connected layers have 256 and 4096 neurons respectively. Their outputs are both followed by a ReLU activation function. Then the generated feature maps are dropped out at the rate of 0.5 only during the training process. The output maps of the last fully connected layer are fed to a softmax function to compute the class scores, which is the last step for CNNs to convert an original image into possibility of classes.

2.2.4. Implementation details

We modify the CNNs architecture similar to AlexNet with batch normalization layer between convolution and activation unit layer. Therefore, the original normalization layers LRN are removed but the dropout [32] layers for fully connected layers are kept.

The CNNs proposed in this paper are implemented in MatConvNet [33] toolbox with a single GPU card of type GeTorce GTX980 and the model is trained for 100 epochs on our image dataset from "Dresden Image Database" [34]. The images from the database are of similar scenes but captured by multiple camera models with multiple devices. Before any further manipulation, the dataset is divided into training and testing set which consisted of 80% and 20% of the samples respectively, and 20% of training samples are randomly selected to form the validation set. Then all the images in these three sets are cropped into 256 × 256 pixel square patches without overlapping and those less than 256 × 256 are ignored. The patches share the same label if they are different parts of one image and they does not experience any kind of color, scale, rotation or aspect ratio augmentation except mean subtraction. Note that the division of training, validation and testing set is done at whole image level in order to avoid the situation that the patches form the same image appear either in training, validation and testing set. Since the results are averaged after running the whole experiment procedure for 5 times, the random split is repeated for 5 times. Before fed into the CNNs, all the patches are transformed into LBP maps by LBP coding operation and then be zero-centered by subtracting the mean of LBP maps of the training set.

We apply mini-batch stochastic gradient descent (SGD) to train our model. A mini-batch size of 100 and the initial learning rate of 0.05 is used. The momentum is fixed to 0.9. The weight-decay is initialized as 0.0005. The parameters in convolution kernels are initialized by random numbers generated from zero-mean Gaussian distribution with standard deviation of 0.01.

The key parameters follow the following decay rule over time: we validate the performance of the trained model every epoch using validation set. If the accuracy of the current validated model is not higher than that of last two validated models, the learning rate would be divided by 2. As a result, we are relieved from manually changing the parameters and the final model could be trained more efficiently until full training process completed and its performance can be better than that of the former validated ones, or not worse at least. So please note, we did not interfere with the training and optimization process of our model after the parameters are initialized and the adjustment policies of parameters are set. Especially, the learning rate is changed automatically under the policy rather than under our subjective supervision.

3. Experimental results and discussion

In this section, we conduct two different experiments to evaluate our proposed method and also provide the results of state-of-the-art classical method [14] and CNN-based method [22] for comparison. We use images from well-known "Dresden Image Database" and cut them into patches as described above. As for the images from various devices of the same model, we only focus on their source camera model and mix them together so that we are able to obtain a relatively larger amount

#	Layer type	Parameter	Value
1	Convolution	Kernel size # of filters Stride Pad	3x3x3 64 2 1
2	Batch Normalization	-	-
3	ReLu	-	-
4	Max pooling	Kernel size Stride Pad	3x3 1 1
5	Convolution	Kernel size # of filters Stride Pad	3x3x64 64 2 1
6	Batch Normalization	-	-
7	ReLu	-	-
8	Max pooling	Kernel size Stride Pad	3x3 1 1
9	Convolution	Kernel size # of filters Stride Pad	3x3x64 32 2 1
10	Batch Normalization	-	-
11	ReLu	-	-
12	Max pooling	Kernel size Stride Pad	3x3 2 0
13	Fully connected	Kernel size # of combinations	31x31x32 256
14	ReLu	-	-
15	Dropout	Dropout rate	0.5
16	Fully connected	Kernel size # of combinations	1x1x256 4096
17	ReLu	-	-
18	Dropout	Dropout rate	0.5
19	Fully connected	Kernel size # of combinations	1x1x4096 Classes
20	Softmax	-	-

Fig. 4. Characteristics and Parameters of our CNNs.

of samples, and that is crucial for CNNs training process. The detailed image lists used in our experiments are shown in Table 1 and some image patches used in the experiments are shown in Fig. 5.

3.1. Experiment 1

Experiment 1 uses the first 12 camera models listed in Table 1. Each patch in our training set is firstly coded based on LBP coding rule as Eqs. (1) and (2), which have been explained in the first subsection of Section 2 . Then the coded images are fed into our CNNs architecture to train the model which is denoted as LBP-CNN. Finally, we use it to match the patches in testing set with their source camera model and calculate the accuracy of this identification process. For a fair comparison, we also train a identification model denoted as EDF (ensemble of demosaicing features) using the method proposed in [14] and report the identification results. Note that the camera models in our experiments are exactly the same with [22] except that we use multiple devices of each model when it is available, and this is a stricter setting than [22] because the training and test samples of the same model in [22] are all form the same device. So it is reasonable to directly reuse the experimental results reported in [22] as a baseline and we denote it as H-CNN.

The classification results of each model of experiment 1 are reported in Table 2. Compared with the accuracy of 94.93% and 98.00% achieved by EDF and H-CNN, LBP-CNN shows an average accuracy of 98.78%, gaining an obvious improvement of 3.85% and 0.78% compared with the baseline. From Fig. 6, we can clearly see that in 8 cases of all 12 camera models, the identification accuracies are higher than H-CNN especially for Praktica DCZ5.9, and the performance of the proposed method outperforms both H-CNN and EDF, which means the proposed method is better form an overall perspective. For Praktica DCZ5.9, LBP-CNN achieves an accuracy of 99.18% while 90.44% of H-CNN, which is

Fig. 5. Some image patches used in the experiments.

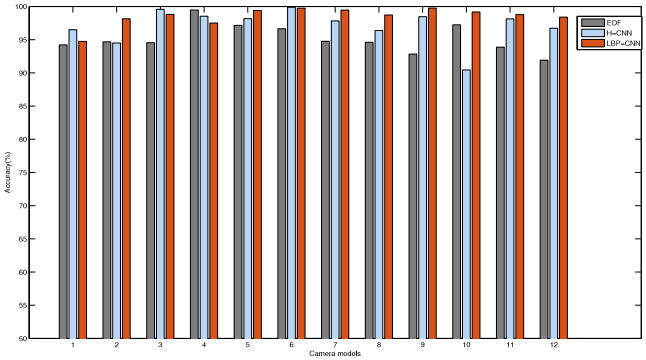

Fig. 6. The identification accuracies of each model in experiment 1.

Table 1
The dataset of camera models from mixed brands. The number of each model corresponds to that of results table.

Seq	Brand	Model	Train	Validation	Test
1	Agfa Photo	DC733s	18 922	4 730	5 832
2	Agfa Photo	DC830i	24 711	6 177	7 668
3	Agfa Photo	Sensor530s	32 000	8 000	10 000
4	Canon	Ixus55	9 688	2 422	3 010
5	Fujifilm	FinePix-J50	32 000	8 000	10 000
6	Kodak	M1063	32 000	8 000	10 000
7	Nikon	D200	32 000	8 000	10 000
8	Olympus	M1050SW	32 000	8 000	10 000
9	Panasonic	DMC-FZ50	32 000	8 000	10 000
10	Praktica	DCZ5.9	32 000	8 000	10 000
11	Samsung	L74wide	32 000	8 000	10 000
12	Samsung	NV15	32 000	8 000	10 000
13	Sony	DSC-H50	32 000	8 000	10 000
14	Sony	DSC-W170	32 000	8 000	10 000
Total			388 291	101 329	126 510

a significant improvement of 8.74%. The least identification accuracy of LBP-CNN is 94.74% recorded by Agfa DC733s while the least accuracy of H-CNN and EDF is 90.44% and 91.92%. There is a clear gap compared with our proposed method. On the other hand, except for Agfa DC733s and Canon Ixus55, the identification accuracies of the left 10 models are all higher than 98%, which is quite encouraging. However, in compared method, there are 6 cases whose identification accuracy is lower than 98%, and this is another proof which illustrates the superiority of the proposed method.

Table 2
The identification accuracies of each model in experiment 1.

Model	EDF	H-CNN	LBP-CNN
Agfa Photo DC733s	94.22%	96.5%	94.74%
Agfa Photo DC830i	94.68%	94.5%	98.15%
Agfa Photo Sensor530s	94.55%	99.57%	98.81%
Canon Ixus55	99.47%	98.54%	97.51%
Fujifilm FinePix-J50	97.16%	98.17%	99.40%
Kodak M1063	96.65%	99.89%	99.75%
Nikon D200	94.76%	97.83%	99.46%
Olympus M1050	94.61%	96.38%	98.72%
Panasonic DMC-FZ50	92.83%	98.46%	99.77%
Praktica DCZ5.9	97.24%	90.44%	99.18%
Samsung L74wide	93.88%	98.13%	98.78%
Samsung NV15	91.92%	96.73%	98.40%
Ave.	94.93%	98.00%	98.78%

Table 3
Average identification accuracy of each method in experiment 2.

Method	EDF	H-CNN	LBP-CN
Ave.	89.12%	97.09%	97.41%

3.2. Experiment 2

The experiment is re-performed on the images from all the 14 camera models listed in Table 1 by adding two camera models Sony DSC-H50 and Sony DSC-W170. It is clearly depicted in Table 3 that the proposed method shows an average accuracy of 97.41%, which is still higher than 97.09% of H-CNN and 89.12% of EDF respectively. The detailed confusion matrix of identification results of LBP-CNN is shown in Table 4. From the confusion matrix, we can see that the highest identification accuracy is 99.85% recorded by Fujifilm FinePix-J50, whereas the lowest identification accuracy is 85.30% recorded by Sony DSC-W170. Except for Sony DSC-H50 and Sony DSC-W170, the identification accuracy of the rest 12 models are all higher than 99%, which proves the effectiveness of the proposed method.

As we can see form Table 4, the decrease of average identification accuracy compared with experiment 1 is mainly caused by the two newly added camera models Sony DSC-H50 and Sony DSC-W170. The identification accuracy of Sony DSC-H50 and Sony DSC-W170 in this experiment is 85.53% and 85.30% respectively, much lower than the averaged identification accuracy. And they are always misclassified into the other class. We also find the similar phenomena in the result of EDF. The cause of this phenomena is that it is difficult to distinguish the images from models of the same brand especially when these models share strong similarity of features [35]. Moreover, in [10], it is reported that LBP is slightly less reliable to identify those two Sony models. This may be the most important reason for the unsatisfactory identification results of Sony DSC-H50 and Sony DSC-W170 in Table 4, because we use the LBP coding as the preprocessing operation of the input images. In spite of this, the average identification accuracy of this method is still the highest among the three methods, which also proves the effective and superiority of the proposed method.

4. Conclusion

In this paper, we investigate a promising approach based on convolutional neural networks with local binary patterns for source camera model identification, which achieves better performance compared to the-state-of-art methods tested on the same dataset. In particular, the LBP coding operation exerts a significant effect on promoting the performance and it indicates that a well-designed CNNs structure with smart preprocessing hint can be an excellent tool for image forensics problems especially for camera identification. In the future, it is prospective for deep learning approaches to perform much better than the classical methods through more careful CNNs structure designing especially the preprocessing hint and better parameter tuning. And we will attempt to investigate smarter hints for CNNs continuously.

Table 4
Confusion matrix of experiment 2 (%). The total accuracy is 94.97%, – means zero or less than 0.1.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Agfa Photo DC733s	99.07	0.34	–	–	–	–	–	–	–	0.27	0.14	–	–	–
Agfa Photo DC830i	0.17	99.62	–	–	–	–	–	–	–	–	–	–	–	–
Agfa Photo Sensor530s	–	–	99.63	–	–	0.13	0.14	–	–	–	–	–	–	–
Canon Ixus55	–	–	–	99.70	–	–	–	–	–	0.17	–	–	–	–
Fujifilm FinePix-J50	–	–	–	–	99.82	–	–	–	–	–	–	–	–	–
Kodak M1063	–	–	–	–	–	99.85	–	–	–	–	–	–	–	–
Nikon D200	–	–	–	–	–	–	99.84	–	–	–	–	–	–	–
Olympus M1050	–	–	–	–	–	0.17	–	99.64	–	–	–	–	–	–
Panasonic DMC-FZ50	–	–	–	–	–	–	–	–	99.78	–	–	–	–	–
Praktica DCZ5.9	–	–	–	0.12	–	–	–	–	–	99.65	–	–	–	–
Samsung L74wide	–	–	–	–	–	–	–	–	–	0.12	99.71	–	–	–
Samsung NV15	0.11	–	–	–	–	0.20	0.10	–	–	–	–	99.43	–	–
Sony DSC-H50	–	–	–	–	–	0.12	–	–	–	–	–	–	85.53	14.26
Sony DSC-W170	–	–	–	–	–	0.12	–	–	–	–	–	–	14.39	85.30

Acknowledgments

This work is supported by the National Science Foundation of China (No. 61502076) and the Scientific Research Project of Liaoning Provincial Education Department, China (No. L2015114).

References

[1] A. Piva, An overview on image forensics, Isrn Signal Process. 2013 (2013).
[2] J. Lukas, J. Fridrich, M. Goljan, Digital camera identification from sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 1 (2) (2006) 205–214. http://dx.doi.org/10.1109/TIFS.2006.873602.
[3] K. San Choi, E.Y. Lam, K.K. Wong, Automatic source camera identification using the intrinsic lens radial distortion, Opt. Express 14 (24) (2006) 11551–11565. http://dx.doi.org/10.1364/OE.14.011551.
[4] A.E. Dirik, H.T. Sencar, N. Memon, Digital single lens reflex camera identification from traces of sensor dust, IEEE Trans. Inf. Forensics Secur. 3 (3) (2008) 539–552. http://dx.doi.org/10.1109/TIFS.2008.926987.
[5] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, A. Piva, Estimate of prnu noise based on different noise models for source camera identification, Int. J. Digit. Crime Forensics 2 (2) (2010) 21–33.
[6] C.T. Li, Source camera identification using enhanced sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 5 (2) (2010) 280–287. http://dx.doi.org/10.1109/TIFS.2010.2046268.
[7] Y. Tomioka, Y. Ito, H. Kitazawa, Robust digital camera identification based on pairwise magnitude relations of clustered sensor pattern noise, IEEE Trans. Inf. Forensics Secur. 8 (12) (2013) 1986–1995. http://dx.doi.org/10.1109/TIFS.2013.2284761.
[8] R. Li, C.T. Li, Y. Guan, Inference of a compact representation of sensor fingerprint for source camera identification, Pattern Recognit. 74 (Supplement C) (2018) 556–567. http://dx.doi.org/10.1016/j.patcog.2017.09.027.
[9] A. Swaminathan, M. Wu, K.R. Liu, Nonintrusive component forensics of visual sensors using output images, IEEE Trans. Inf. Forensics Secur. 2 (1) (2007) 91–106. http://dx.doi.org/10.1109/TIFS.2006.890307.
[10] G. Xu, Y.Q. Shi, Camera model identification using local binary patterns, in: Proceedings of IEEE International Conference on Multimedia and Expo, IEEE, 2012, pp. 392–397. http://dx.doi.org/10.1109/ICME.2012.87.
[11] Y. Hu, C.T. Li, X. Lin, B.b. Liu, An improved algorithm for camera model identification using inter-channel demosaicking traces, in: Proceedings of 20th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2012, pp. 325–330. http://dx.doi.org/10.1109/IIH-MSP.2012.85.
[12] A. Tuama, F. Comby, M. Chaumont, Camera model identification based machine learning approach with high order statistics features, in: Proceedings of the 24th European Signal Processing Conference, 2016, pp. 1183–1187. http://dx.doi.org/10.1109/EUSIPCO.2016.7760435.
[13] B. Xu, X. Wang, X. Zhou, J. Xi, S. Wang, Source camera identification from image texture features, Neurocomputing 207 (2016) 131–140. http://dx.doi.org/10.1016/j.neucom.2016.05.012.
[14] C. Chen, M.C. Stamm, Camera model identification framework using an ensemble of demosaicing features, in: 2015 IEEE International Workshop on Information Forensics and Security (WIFS), 2015, pp. 1–6. http://dx.doi.org/10.1109/WIFS.2015.7368573.
[15] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.
[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9. http://dx.doi.org/10.1109/CVPR.2015.7298594.
[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.
[19] L. Bondi, L. Baroffio, D. Gera, P. Bestagini, E.J. Delp, S. Tubaro, First steps toward camera model identification with convolutional neural networks, IEEE Signal Process. Lett. 24 (3) (2017) 259–263. http://dx.doi.org/10.1109/LSP.2016.2641006.
[20] J. Chen, X. Kang, Y. Liu, Z.J. Wang, Median filtering forensics based on convolutional neural networks, IEEE Signal Process. Lett. 22 (11) (2015) 1849–1853. http://dx.doi.org/10.1109/LSP.2015.2438008.
[21] B. Bayar, M.C. Stamm, A deep learning approach to universal image manipulation detection using a new convolutional layer, in: Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, ACM, 2016, pp. 5–10. http://dx.doi.org/10.1145/2909827.2930786.
[22] A. Tuama, F. Comby, M. Chaumont, Camera model identification with the use of deep convolutional neural networks, in: Proceedings of IEEE International Workshop on Information Forensics and Security, 2016, pp. 6–pages. http://dx.doi.org/10.1109/WIFS.2016.7823908.
[23] Y. Qian, J. Dong, W. Wang, T. Tan, Deep learning for steganalysis via convolutional neural networks, in: Proceedings of SPIE Electronic Imaging, International Society for Optics and Photonics, 2015. http://dx.doi.org/10.1117/12.2083479. 94090J–94090J.
[24] Y. Chen, Y. Huang, X. Ding, Camera model identification with residual neural network, in: 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 4337–4341. http://dx.doi.org/10.1109/ICIP.2017.8297101.
[25] Source camera identification based on content-adaptive fusion residual networks, Pattern Recognit. Lett. (2017). http://dx.doi.org/10.1016/j.patrec.2017.10.016.
[26] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognit. 29 (1) (1996) 51–59. http://dx.doi.org/10.1016/0031-3203(95)00067-4.
[27] M. Pietikäinen, Local binary patterns, Scholarpedia 5 (3) (2010) 9775. http://dx.doi.org/10.4249/scholarpedia.9775.
[28] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 807–814.
[29] Y.-L. Boureau, J. Ponce, Y. LeCun, A theoretical analysis of feature pooling in visual recognition, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 111–118.
[30] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 37, PMLR, Lille, France, 2015, pp. 448–456.
[31] G. Xu, H.-Z. Wu, Y.-Q. Shi, Structural design of convolutional neural networks for steganalysis, IEEE Signal Process. Lett. 23 (5) (2016) 708–712. http://dx.doi.org/10.1109/LSP.2016.2548421.
[32] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[33] A. Vedaldi, K. Lenc, Matconvnet: Convolutional neural networks for matlab, in: Proceedings of the 23rd ACM International Conference on Multimedia, ACM, 2015, pp. 689–692.
[34] T. Gloe, R. Böhme, The dresden image database for benchmarking digital image forensics, J. Digit. Forensic Pract. 3 (2–4) (2010) 150–159. http://dx.doi.org/10.1080/15567281.2010.531500.
[35] M. Kirchner, T. Gloe, Forensic camera model identification, in: Handbook of Digital Forensics of Multimedia Data and Devices, 2015, pp. 329–374. http://dx.doi.org/10.1002/9781118705773.ch9.