IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 8, 2019, accepted December 3, 2019, date of publication December 13, 2019, date of current version March 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2959627

Source Camera Identification Based on Coupling Coding and Adaptive Filter

MENGNAN ZHAO[1], (Student Member, IEEE), BO WANG[1], (Member, IEEE), FEI WEI[2], (Member, IEEE), MEINENG ZHU[3], (Member, IEEE), AND XUE SUI[4]

[1]School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, China
[2]Department of Electrical Engineering, The State University of New York at Buffalo, Buffalo, NY 14260-2500, USA
[3]Beijing Institute of Electronics Technology and Application, Beijing 100091, China
[4]College of Psychology, Liaoning Normal University, Dalian 116029, China

Corresponding author: Bo Wang (bowang@dlut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant U1936117 and Grant 61772111.

ABSTRACT Source Camera Identification (SCI) has been playing an important role in the security field for decades. With the development of Deep Learning, the performance of SCI has been noteworthily improved. However, most of the proposed methods are forensic only for a single camera identification category, e.g., the camera model identification. For exploiting the coupling between different camera categories, we present a new coding method. That is, we apply the multi-task training method to regress the categories, namely, to classify brands, models and devices synchronously in a single network. Different from the common multi-task method, we obtain the multi-class classification result by just one single label classification. To be specific, we classify the categories in a progressive way that the parent category classification result will be used in the child category classification (a detailed explanation will be given later in the main context). Also, by appropriately increasing the redundancy of the coding method for classifying new camera categories, the training time can be greatly reduced. To better extract camera attributes, we propose an adaptive filter. Additionally, we propose an auxiliary classifier that only focuses on the camera model re-classification, due to the low performance of the main classifier on certain models. Lastly, the extensive experiments show that our methods have a better performance than other existing methods.

INDEX TERMS Source camera identification, deep learning, multi-task training, camera categories coupling coding, adaptive filter, auxiliary classifier.

I. INTRODUCTION

With the rapid development of multimedia technology, digital images have gained growing popularity on the idea expressing. In many scenarios, digital images are playing more important roles [1], [2]. For example, they can be treated as evidences in criminal investigations. However, as the image editing applications are speedily evolving, modifying digital images is no longer the job that needs professional skills [3]–[6]. That can cause a series of problems. Like the example we mentioned before, it may affect justice and the law enforcement. In order to ensure the credibility of the image, source identification on digital images become necessary. Many forensics algorithms have been proposed to identify the source of digital image [7], [8]. The essence of camera forensics is to detect the camera attribute difference. As shown in Fig.1, there are multiple steps in the processing between a real image to a digital image. For each step of the imaging processing, there are corresponding traditional methods proposed to classify the images [9], [10].

In the past few years, SCI performance has been greatly improved. By the powerful learning capability, CNN (Convolutional Neural Network) can automatically learn the differences among different cameras [11], [12]. The network performance highly depends on the number of images in the training set. Increasing the number of training images will improve the accuracy but the training time will also be increased. Limited by storage capacity of hardware devices, original images taken by the camera are difficult to be directly used as the input of CNN, which will generate excessive parameters. Therefore, the existing method based on deep learning divides the images into fixed-size blocks [13].

The associate editor coordinating the review of this manuscript and approving it for publication was Kim-Kwang Raymond Choo.

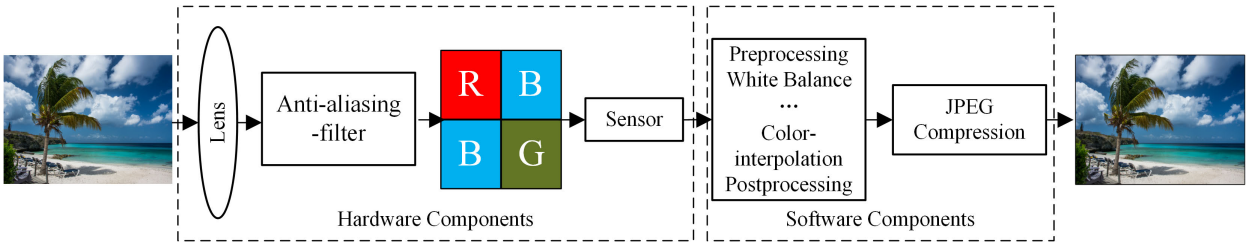


FIGURE 1. Image processing pipeline.

SCI includes the identifications on the brand, model and device. Although the proposed camera forensics methods improve significantly, there are still issues need to be resolved. Firstly, most of the existing methods only focus on a single category, such as the camera model. Although the camera brand can be identified with high accuracy, there are still room for the model and device classification accuracy to be improved. For the SCI, all existing methods did not put the correlation among the three categories into consideration. Then, the classification network need to be retrained for classifying the new categories, even if there are only few catregories. Also, the performance will be greatly affected by the image contents while extracting the identification attributes of the camera. Therefore, it is necessary to preprocess the images. Tuama *et al.* [14] use a high-pass filter and a wavelet-based denoising filter to remove the image contents. However, using a fixed high-pass filter may remove the original camera attributes as well.

In our work, we propose the category coupling multi-task training method based on the adaptive filter. In order to make full use of the correlation between camera categories of brands, models and devices, we adopt a progressive method to classify the three categories. We start with the classification on camera brands and categorize images into different visual subspaces (brands). The model classification is done separately in each corresponding brand subspace. Meanwhile, we expect that the classification on subclasses can in turn improve the classification accuracy of the parent class. In this way, we consider to use the single label to realize the multi-class classification, which called as *SLMC*. Besides, we set redundancy coding to improve the scalability of the network before training, which can reduce the network training time for new categories. For a better image attribute information extracting, we use the residual learning to extract image contents with multi-layer convolution. By concatenating the output of each layer of the convolution kernel, 1×1 convolution kernel is used to selectively extract the low-frequency or high-frequency contents relevant to SCI. For the local neighborhood differences of the camera lens, we train an additional position classifier as an auxiliary classifier to reclassify some camera categories.

II. RELATED WORKS

In general, a digital camera consists of two major subsystems, the hardware and software [15]. The most common forensics methods contains hardware part are based on the optical aberration [9] and the Sensor Pattern Noise (SPN) [16], [17]. Similarity, the software part involve JPEG compression [10] and color interpolation [18]. The light alters dramatically in real environment, so it is improper to use illuminance as a SCI feature. However, the illuminance is consistent in the same image. Riess *et al.* [19] used illuminance as the feature of image forgery detection. As we know, cameras from different manufacturers have differences in lens distortion parameters, such that the interpolation map for specific camera lens distortion can be considered as fixed. Therefore, Hwang *et al.* [20] used interpolation based lens distortion parameters as a feature to classify the model of camera.

The difference of sensor SPN makes it becoming the most widely studied camera forensics method and the distortion introduced by SPN is very helpful for camera model classification. SPN includes fixed pattern noise and Pattern Responding Non-Uniform noise (PRNU). The PRNU is generated by the non-uniformity of the hardware sensitivity to different illuminance intensity. Lucas *et al.* [21] obtained camera fingerprints by extracting the average residuals in amounts of images. Through the decomposition and combination of image color channels, Li *et al.* [22] obtained relatively complete PRNU, they propose the DPRNU method to verify the integrity of camera images. By calculating the relationship between neighboring pixels of different color channels, Choi *et al.* [23] proposed a method based on color interpolation to capture camera categories. Most cameras use JPEG compression to store the final image.

For different cameras, the size of the image is different, so the quality of the image produced by different cameras is also quite different. Choi classified images by JPEG compression for the first time. Similar to Riess, Mahdian *et al.* [24] used JPEG compression to detect forgery of images. Farah Ahmed *et al.* [25] proposed a comparative analysis of SCI between deep learning and traditional methods (PRNU). Camera forgery methods include Seam Carving, Fingerprint Copying, and Adaptive PRNU Denoising, etc.

Sameer *et al.* [26] proposed a method based on deep learning to detect camera forgery. Bondi *et al.* [27] proposed the method which combines CNN and SVM classifier. It uses CNN to extract features, and then uses SVM classifier for identification. At the same time, with the popularity of mobile phones, mobile device-based source

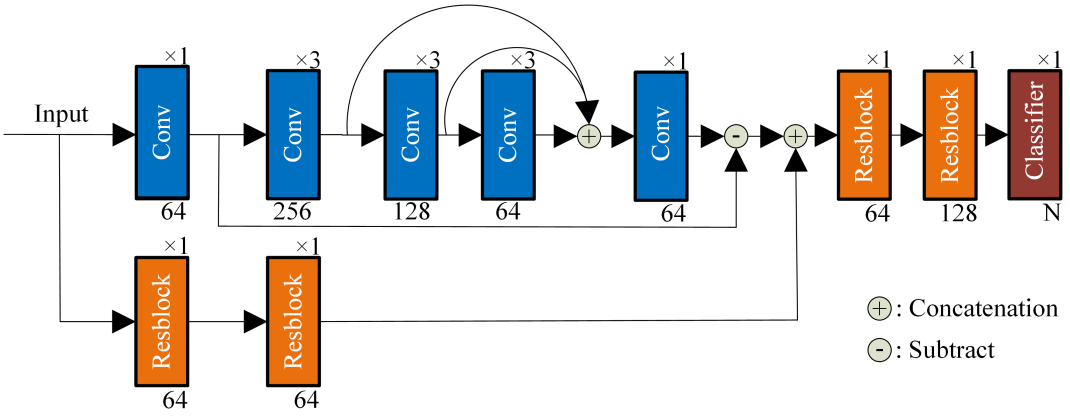


FIGURE 2. The architecture of the proposed network for SCI. Where $\times i$ denotes the repeat times and *Classifiers* can be found in Fig. 3.

camera forensics [28]–[30] are becoming more and more extensive.

No matter what been used is a deep learning method or a traditional method, the camera attribute extraction is affected by image contents and various noises. Therefore, Tuama *et al.* [14] proposed to pre-process the input image by extracting the high-frequency texture using a low-pass filter, and then classifying the high-frequency image. Bayar *et al.* [11] proposed a robust CNN-based camera model identification. By using the constrained convolution layer, camera model identification is robust to re-compression.

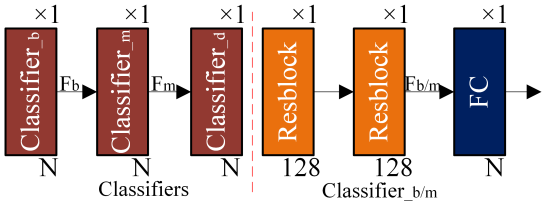


FIGURE 3. Proposed camera attributes classifier.

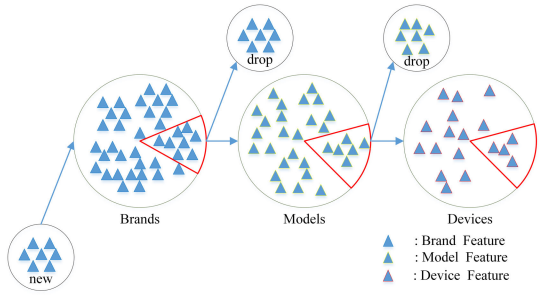


FIGURE 4. Recursive method to extract camera features.

III. PROPOSED METHODS

In this section, we first describe the proposed network architectures. We analyze network from the perspective of residual learning and use a new coding method which is different from other methods. Noteworthy, our coding method is easily transferred to other similar tasks. In what follows, we describe our method with details.

A. OVERALL STRUCTURE

As shown in Fig.2, we use the method of residual learning [13], [31] to extract the image contents. Firstly, we use one convolution layer to extract the image features and then use the multi-layer convolution to extract the contents of the image. Next, the output of the multi-layer convolution kernel is concatenated to extract the contents that have the same dimension with image features through the 1×1 convolution kernel. CNN can determine what needs to be removed in order to maximize the preserve information which is related to the classification attributes. The identification method removes the contents that disturbs the classification, and preserve camera attributes as much as possible. However, the residual learning network disrupts the correlation between original camera image neighborhoods. Therefore, we use a series of convolution layers to extract the relevant information of the image neighborhoods separately, and then concatenate the two parts of information together as the final output feature.

This paper uses the *SLMC* method to classify the camera categories. Different from the previous classification method, when extracting image features, we want to extract enough common features that can simultaneously classify camera brands, models, and devices. As shown in Fig.4, we use recursive method to extract camera features. The sub-classifier can affect the parent-classifier to drop some features that are invalid for sub-classification. Finally, we extract enough common features to classify the three categories.

Based on this extracting process, we propose a new coding method. As shown in Table 1, the existing deep learning methods classify brands, models, and devices with output categories of 14, 27, and 74, respectively. All methods do not consider the correlation among the three categories of a

TABLE 1. Comparison of bit number in proposed coding (PC) method and original coding (OC) method.

	Brands(BL)	Models(ML)	Devices(DL)	Binary bits
OC	14	27	74	115
PC	14	5	5	24

TABLE 2. Coding method for *Sony_DSC*.

	Brands	Models	Devices
Sony_DSC-H50_0	1	10	100
Sony_DSC-H50_1	1	10	010
Sony_DSC-H50_2	1	10	001
Sony_DSC-W170_0	1	01	100
Sony_DSC-W170_1	1	01	010

single camera. The multi-classification method will impact the classification performance of the network. The more output categories, the more significant impact there will be. The performance of binary-classifier is usually better than the multi-classifier. In order to improve the classification performance for models and devices, we choose to classify models (devices) respectively under the same brands (models), which transforms the unconstrained multi-classification into the constrained binary or trinary-classification. As shown in Table 1, the encoding length of our coding method is

$$N = b_l + \max(m_l[i]) + \max(d_l[j]), \tag{1}$$

where $i = 0, 1, \ldots, b_l - 1$ and $j = 0, 1, \ldots, m_l - 1$. N is the encoding length, b_l denotes the number of brands, m_l and d_l are the number of models and devices under the same parent-class, respectively.

We first select some images (*Sony_DSC*) as the pre-training data set. The encoding method is shown in Table 2. Since all camera devices are of the same brand, there is no need to classify the brands. We use six-bits to represent output categories. For example, the ideal result of the $Sony\_DSC - H50\_0$ camera model classifier output is 110000, and the camera devices classifier output is 110100, which is classified in a progressive way. We denote the category of the devices or models which is less than the binary bits (e.g., the number of devices of $Sony\_DSC - W170$ is less than 3) as the coding redundancy. For example, there are no devices encoded as 101001.

Coding redundancy will impact network performance. However, experiments show that such impact on performance can be neglected (cannot be neglected when setting too many coding redundancy), and the coding between different classes does not affect each other. That is, when we classify the devices, the binary bits of the brand and model will not be set. The details will be given in experiment section.

The process is given in Algorithm 1. Where Classify denotes softmax cross entropy function. Conv3 and Conv1 represent the 3×3 convolution and 1×1 convolution operation. However, this approach does not increase coupling

Algorithm 1 Training Algorithm for Proposed Network

Input: x: image(64×64); Label: Ground truth label; *B_label*: Brand index ([1,. . .,1,0,. . .,0,0,. . .,0]); *M_label*: Model index ([0,. . .,0,1,. . .,1,0,. . .,0]); *D_label*: Device index ([0,. . .,0,0,. . .,0,1,. . .,1])

Output: loss *Cost_b*, *Cost_m*, *Cost_d*

1: extract image features: Fea = $Conv3(x)$
2: extract image contents C: C_1,C_2,C_3 = $Conv3$(Fea); C = $Conv1$(Concat([C_1,C_2,C_3]))
3: obtain the attributes of camera: Att = Fea - C; $Conv_1$,$Conv_2$,$Conv_3$ = $Conv$(Att)
4: compute brand label: Label1 = $abs(FC(Conv_1))$
5: compute model label: Label2 = $abs($ $FC(Conv_2)$ - Label1)
6: compute device label: Label3 = $abs(FC(Conv_3)$ - Label2 - Label1)
7: *Cost_b* = Classify((Label1), Label & *B_label*)
8: *Cost_m* = Classify((Label2), Label & *M_label*)
9: *Cost_d* = Classify((Label3), Label & *D_label*)

TABLE 3. The network divides the three categories to different numerical spaces to eliminate categories coupling. [a,b] denotes the arbitrary range of values which are obtained from FC(Conv1). k and n (where 0 << k << n) represent the order of magnitude.

	Brands	Models	Devices
FC(Conv1)	$[a,b]$	0	0
Softmax	1	0	0
FC(Conv2)	$[a,b]$	$10^k \times [a,b]$	0
Softmax	0	1	0
FC(Conv3)	$[a,b]$	$10^k \times [a,b]$	$10^n \times [a,b]$
Softmax	0	0	1

among different camera categories. For example, we assume that the *label* of given images is 110100, and the camera brand (1-bit), model (2-bits), device (3-bits) are encoded as 100000, 010000, and 000100, respectively. Ideally, the output by the device classifier is consistent with the *label* (actually 110100). However, models may divide different camera categories into different numerical spaces and indirectly eliminate the coupling between categories. Then coding method degenerates into a separate classification. As shown in Table 3, the division of the numerical spaces make the parent class (brand, model) having no effect on the loss of the subclass (model, device). That is, the parent bit (arbitrary value that range in [a, b]) in the subclass has no effect on the subclass classifier after the softmax function. So we propose the method of correlation loss.

B. LOSS FUNCTION

SLMC is a progressive method when classify the camera attributes. The model classification works under the condition that the brand classification is accurate. Likewise, device classifier can be better trained whereas the camera brands and model classification are accurately classified. As shown

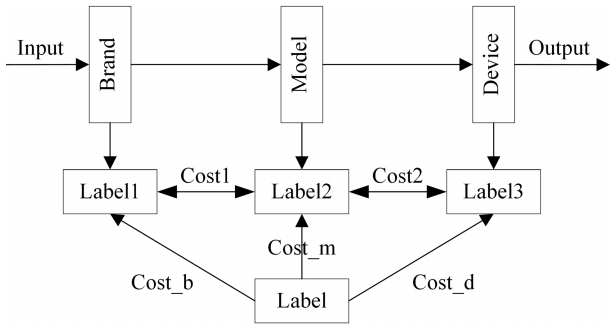


FIGURE 5. Improved framework for classification methods.

in Table 1, the camera brands occupy most of binary bits, which results in the lowest accuracy on the camera brand classification when network randomly initializes the weights. As described in the previous section, this training method is pathological. Therefore, in order to prevent the cameras classification from falling into local minimum points and solve the problem of pathological, when classifying the camera models or devices, we can replace the formula in Algorithm 1 with the following formula

$$Label1 = 1 \times Softmax(FC(Conv\_1)) \quad (2)$$

$$Label2 = 2 \times Softmax(FC(Conv\_2)) \quad (3)$$

$$Label3 = 3 \times Softmax(FC(Conv\_3)) \quad (4)$$

where $Label1, Lael2-Label1$ and $Label3-label2$ can represent the output logits of the network for camera brand, model and device. However, this method just solves the problem of dividing numerical spaces, sub-classes do not improve the classification accuracy of the parent class. We further modify the loss function, as shown in Fig.5. $Cost\_b$, $Cost\_m$ and $Cost\_d$ are defined in Algorithm 2. $Cost1$ and $Cost2$ are used to assist the information sharing among the three categories, which are defined as the following

$$Cost1 = L_1(Label2[: b_l], Label1[: b_l]) \quad (5)$$

$$Cost2 = L_1(Label3[: (b_l + m_l)], Label2[: (b_l + m_l)]) \quad (6)$$

where $Label$ has three bits been set to 1. L_1 stands for L_1 norm [32]. b_l and m_l denote the binary bits of brand and model. Therefore, for the devices classification, they classify the camera brands and models as well. However, this method has the same loss on the three categories of cameras implying that additional weight settings and progressive training are still necessary.

$$Cost = \alpha \times Cost\_b + \beta \times Cost\_m + \theta \times Cost\_d + \mu \times Cost1 + \nu \times Cost2 \quad (7)$$

We recommend to set the above parameters to 0.5, 0.4, 0.1, 0.1 and 0.1, respectively.

C. CODING REDUNDANCY

For the proposed methods, the number of final outputs obtained by the network is fixed. Accordingly, the model needs to be retrained for classifying the new data. In order to improve the scalability of the network, it is appropriate to set redundancy for network classification before the training. Extended classifier has impacts on the model performance, but it increases the ability of network retraining.

As shown in Table 4, we give the number of binary bits of our proposed methods and existing classification methods. When new data needs to be classified, the trained model can be used as a pre-training model, which can greatly reduce the training time. We define the representation of redundancy as

$$Red = (C_b \times C_m \times C_d - OC_d)/(C_b + C_m + C_d) \quad (8)$$

where C_b, C_m, C_d represent the number of category binary bits in the corresponding classifier. We denote OC_d as the amount of devices in training set. Redundancy can simply measure the efficiency of the coding. Simultaneously, high value of redundancy will affect the performance of the classification network. The higher the value of redundancy, the greater the impact on the classification network performance.

Algorithm 2 Improved Algorithm for Classification

Input: Label: Ground truth label; *B_label*: Brand index ([1,. . .,1,0,. . .,0,0,. . .,0]); *M_label*: Model index ([1,. . .,1,1,. . .,1,0,. . .,0]); *D_label*: Device index ([1,. . .,1,1,. . .,1,1,. . .,1])

Output: loss *Cost_b*, *Cost_m*, *Cost_d*

1: compute brand label: Label1 = $Softmax(FC(Conv_1))$
2: compute model label: Label2 = $2 \cdot Softmax(FC(Conv_2))$
3: compute device label: Label3 = $3 \cdot Softmax(FC(Conv_3))$
4: *Classify2*(logits,label) = $-Sum(\text{label} \times \log(\text{logits}))$
5: *Cost_b* = *Classify2*((Label1), Label & *B_label*)
6: *Cost_m* = *Classify2*((Label2), Label & *M_label*)
7: *Cost_d* = *Classify2*((Label3), Label & *D_label*)

TABLE 4. The proposed redundant coding. Which can reduce the training time of the network for new camera categories.

	Brands	Models	Devices	Binary bits	Redundancy
Original-Classifier(OC)	14	27	74	115	-
Extended-Original-Classifier(EOC)	15	90	540	645	-
Classifier(CL)	14	5	5	24	9.792
Extended-Classifier(EC)	15	6	6	27	19.63

D. AUXILIARY CLASSIFIER

The proposed methods work well for most models, but for some models, e.g., the D70 and the D70s, the classification performance is not as good as what is expected. Therefore, we propose an auxiliary classifier to improve the classification performance of the main classifier. However, this classification method requires a separate classifier for each camera model, namely, the requirement on memory will be higher. Therefore, we only use the classifier as an auxiliary

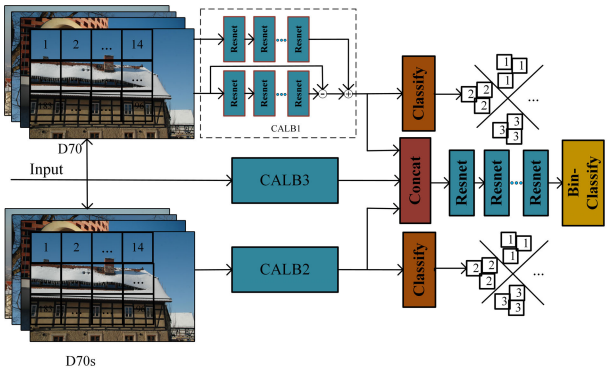


FIGURE 6. Proposed camera attributes auxiliary classifier.

classifier to re-classify camera categories which are difficult to be classified by the main classifier.

As shown in Fig.6, the same position of different pictures (for the same camera model) divide into the same class, i.e., the classification based on the lens position. Too many patches in the same image will reduce the accuracy of the Position Classifier (PC). In our experiment, we choose the patch size to be 48. Considering the difference on camera category sizes, we only use the top left corner of the image for classification. The selected area is

$$input = img[patch\_size \times C_{line}, patch\_size \times C_{Col}] \quad (9)$$

Both C_{line} and C_{Col} are set to 14, which means that images are divided into 196 categories. By the large coupling between neighbor pixels and the limitation of training data, the classification performance of the PC is poor. The coupling between adjacent categories can be reduced by setting stride appropriately when dividing patches. We take the value of stride to 30 and retrain the network for the comparing experiments. Fortunately, despite the bad classification performance of the PC, the extracted position information still contributes to the Binary Classifier (BC).

For any camera category, we first train the PC. The network can better extract the camera attribute information when the PC gradually improves the classification accuracy. It is worth noting that the neighborhood classification method can extract attribute information of any local position of the camera. However, when the image from the same camera passes the same post-processing method (different processing methods for different camera categories), the location-based classifier may ignore the global information. Therefore, with the multi-location classifier, we set additional networks to extract global information from different camera models. We fix the network parameters of PC when we train the BC. The input of the BC is the concatenate of three network outputs. More details are given in Algorithm 3.

Algorithm 3 Training Algorithm for Auxiliary Classifier

Input: x_1: blocks (48 × 48×3) from D70; $Label\_1$ = [1,0]; x_2: blocks (48 × 48×3) from D70s; $Label\_2$ = [0,1]; $PClabel_1$: position label for x_1; $PClabel_2$: position label for x_2; Input: x_1 or x_2

Output: classify images from D70 and D70s

1: train position classifier 1: $Fea\_1 = CALB1(x_1)$; min[$loss(PC(Fea\_1), PClabel_1)$];
2: train position classifier 2: $Fea\_1 = CALB2(x_2)$; min[$loss(PC(Fea\_2), PClabel_2)$];
3: global feature extract: $Fea\_3 = CALB3$(Input);
4: train binary classifier: $Fea\_1 = CALB1$(Input); $Fea\_2 = CALB2$(Input); Fea = Concat[Fea_1, Fea_2, Fea_3]; min[$loss(BC$(Fea),Label)]

IV. EXPERIMENT

A. DATASETS

In our experiments, all data comes from the publicly available Dresden database [33]. We select all 74 categories to evaluate the network performance. Same as the experimental settings of [13], we divide the dataset into training and testing sets randomly, where 70% of the data is chosen for training and the rest 30% is for the testing data. Network performance is evaluated by calculating the average accuracy of all image blocks from the testing set.

TABLE 5. Pre-training network analysis of redundant coding.

Brand	Model	Device				
00000000000001	-	-10000	-01000	-00100	-00010	-00001
Sony_DSC-T77_0	00001	0.714	0.0976	0.1874	<0.0001	<0.0001
Sony_DSC-T77_1		0.103	0.639	0.2564	<0.0001	<0.0001
Sony_DSC-T77_2		0.126	0.1348	0.738	<0.0001	<0.0001
Sony_DSC-W170_0	00010	0.635	0.365	<0.0001	<0.0001	<0.0001
Sony_DSC-W170_1		0.391	0.6093	<0.0001	<0.0001	<0.0001

B. PRE-TRIANING

We first pre-train the network with images from *Sony_DSC*. The pre-training is used to test the effect of redundant coding on classification performance. For saving the training time, we crop the image before training. All input images are clipped to 48 × 48 blocks with the non-overlapping method. Considering the limit of hardware storage capacity, we set the batch to 96.

We use the Extended-Classifier to encode the camera categories. As shown in Table 5, when we train the network with *Sony_DSC*, the device coding bits will generate redundancy, but it has small effect (<0.0001) on the device classification performance. Therefore, the new encoding method can be well applied to classify camera categories.

C. EVALUATION ON FINAL DATASET

Now, we train the network with 74 devices from 14 brands. Same as the experiment settings above, we first crop the original image to 48 × 48 blocks. If the size of image is 2500 × 2000 (may be 2560 × 1920 and other shapes), which means that every image contains an average of 2,000 blocks. It takes too much time to use all the blocks as the training

TABLE 6. Confusion matrix for 27 camera models using the proposed method. Results are obtained by average all camera accuracy in the same model.

	Agfa_DC_504	Agfa_DC_733s	Agfa_DC_830i	Agfa_Sensor505x	Agfa_Sensor530s	Cannon_Ixus55	Cannon_Ixus70	Cannon_PowerShotA640	Casio_EX_Z150	FujiFilm_FinePixJ50	Kodak_M1063	Nikon_CookPixS710	Nikon_D200	Nikon_D70	Nikon_D70s	Olympus_mju_1050SW	Panasonic_DMC_FZ50	Pentax_OptioA40	PentaxOptioW60	Praktica_DCZ5.9	Picoh_GX100	Rollei_RCP-7325XS	Samsung_L74wide	Samsung_NV15	Sony_DSC_H50	Sony_DSC_T77	Sony_DSC_W170
Agfa_DC_504	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Agfa_DC_733s	0.00	0.98	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Agfa_DC_830i	0.01	0.03	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Agfa_Sensor505x	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
Agfa_Sensor530s	0.00	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
Cannon_Ixus55	0.00	0.00	0.00	0.00	0.00	0.90	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
Cannon_Ixus70	0.00	0.00	0.00	0.00	0.00	0.04	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Cannon_PowerShotA640	0.00	0.00	0.00	0.00	0.00	0.02	0.06	0.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Casio_EX_Z150	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FujiFilm_FinePixJ50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kodak_M1063	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon_CookPixS710	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon_D200	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon_D70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.64	0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon_D70s	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.41	0.58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Olympus_mju_1050SW	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Panasonic_DMC_FZ50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Pentax_OptioA40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PentaxOptioW60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Praktica_DCZ5.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Picoh_GX100	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Rollei_RCP-7325XS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Samsung_L74wide	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.01	0.00	0.00	0.00
Samsung_NV15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.40	0.02	0.98	0.00	0.00	0.00
Sony_DSC_H50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.04	0.02
Sony_DSC_T77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.97	0.01
Sony_DSC_W170	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.02	0.87

set. So we randomly extract 60000 images from training set to train the network and re-produce the training set using the same way every epoch. We reset the learning rate to 0.001 and change it with the training process. The final results are shown in Table 7. We implement the proposed network structure by using Tensorflow®. All experiments are trained with NVIDIA GTX 1080Ti GPUs. When testing the camera models, we classify all devices under the same model into the same class.

Table 10 shows the confusion matrix of brands. Table 6 shows the confusion matrix of models. Table 11 shows several device classification accuracy of the data set used in the experiment. Camera models can be classified by model classifier and device classifier. Table 7 shows that the performance of the model classifier is better than the device classifier using our proposed method. Meanwhile, our method performs better than the existing methods no matter the model classifier and the device classifier.

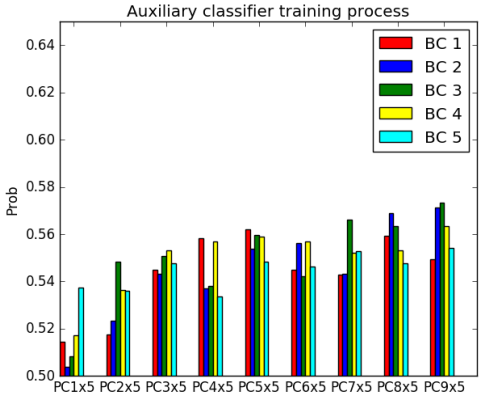


FIGURE 7. Accuracy of training process of auxiliary classifier (PC+BC).

As shown in Table 6, the proposed classifier has low classification performance for the Nikon D70 and the Nikon D70s. This phenomenon is not only presented in our

TABLE 7. Camera identification accuracy compared with the previous methods. Where *W/OCouple + Adap* stands for without couple coding and adaptive filter and *W/OCouple* stands for without couple coding. Two ablation experiments just train for camera models.

	Brand	Model	Device
CNN[28]	0.972	0.916	0.305
RESNET[33]	0.978	0.943	0.458
W/O Couple+Adap	-	0.932	-
W/O Couple	-	0.948	-
Brand Classifier	**0.994**	-	-
Model Classifier	**0.990**	**0.961**	-
Device Classifier	0.987	0.954	**0.475**

TABLE 8. Ablation study of position classifier (PC) and global feature extract (GFE) in auxiliary classifier.

	Origin	GFE+BC	PC+BC	PC+BC+GFE	PC+BC+GFE+stride
Accuracy	0.580	0.566	0.571	0.593	0.605

proposed method. Therefore, a reasonable explanation is that two camera models have high similarity. We classify all devices belonging to the same camera model into same categories. The blocks corresponding to the same position in different images are classified into the same class. Each image is divided into 196 blocks, namely, 196 categories. In order to better train the BC, we need to extract enough camera attributes. The classification performance of the PC is positively correlated with the features (position attributes) extracted by the classifier. During the training, limited size of data and neighborhood similarity make the classification performance of the network greatly reduced. Over-fitting occurs when there are too many training times. At the same time, the attribute information extracted by the PC may not always be helpful to BC, although it can effectively distinguish different positions of the images.

As shown in Algorithm 3, we use the alternating training method to gradually train the entire network. We choose to change the network every five epochs. Our final results are shown in Table 8. Comparing with the original results, the effect of PC is not obvious. BC classification performance is subject to the performance of PC classification, whereas PC classification performance is bad (about 0.1, random guess accuracy is 1/196). However, we believe that with the size of training data set growing, the performance of the PC will gradually improve. In order to show the impact of PC on BC, we removed the global information extraction network and used the location information extracted by PC to classify BC. As shown in Fig.7, each alternation includes 5 epochs (for BC training). We provide 45 epoch results, and further training cannot get further improvement in the network performance.

V. THE SCALABILITY

In this section, we test the scalability of the proposed method. The selected categories are shown in the Table.9.

TABLE 9. The camera categories used to detect the performance of network scalability.

Canon_Power ShotA640	Samsung_NV 15_0	Nikon_D200	**Agfa_Sensor 505-x_0**
Canon_Ixus 70_0	Samsung_NV 15_1	**Nikon_D70**	-
Canon_Ixus 70_1	**Samsung_N V15_2**	-	-
Canon_Ixus 70_2	-	-	-

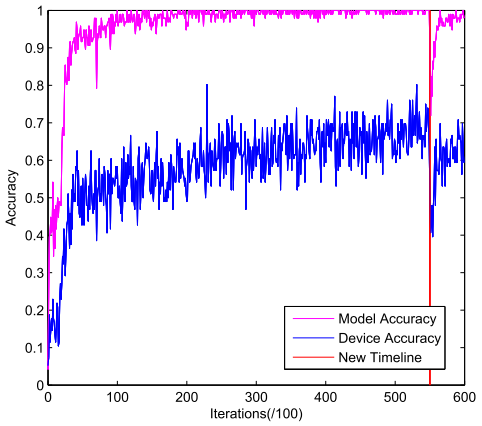


FIGURE 8. The accuracy curve to show the efficiency of the network scalability.

TABLE 10. Confusion matrix for 14 camera brands using the proposed method. Results are obtained by average all camera accuracy in the same brand.

	Agfa	Cannon	Casio	FujiFilm	Kocak	Nikon	Olympus	Panasonic	Pentax	Praktica	Ricoh	Rolli	Samsung	Sony
Agfa	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Cannon	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00
Casio	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FujiFilm	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kocak	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Nikon	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Olympus	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Panasonic	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Pentax	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Praktica	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Ricoh	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Rolli	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Samsung	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Sony	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Samsung_NV15_2, Nikon_D70 and Agfa_Sensor505-x_0 are adopted to detect the network scalability for camera device, model and brand. We train the network under the same coding methods and length as the experiment settings of the paper. The samples of black fonts in the Table.9 are

TABLE 11. Classification accuracy of several devices.

	Samsung_NV15_0	Samsung_NV15_1	Samsung_NV15_2	Casio_EX_Z150_0	Casio_EX_Z150_1	Casio_EX_Z150_2	Sony_DSC_T77_0	Sony_DSC_T77_1	Sony_DSC_T77_2
Samsung_NV15_0	0.57	0.26	0.16	0.00	0.00	0.00	0.00	0.00	0.00
Samsung_NV15_1	0.33	0.46	0.21	0.00	0.00	0.00	0.00	0.00	0.00
Samsung_NV15_2	0.22	0.40	0.38	0.00	0.00	0.00	0.00	0.00	0.00
Casio_EX_Z150_0	0.00	0.00	0.00	0.50	0.31	0.19	0.00	0.00	0.00
Casio_EX_Z150_1	0.00	0.00	0.00	0.47	0.28	0.25	0.00	0.00	0.00
Casio_EX_Z150_2	0.00	0.00	0.00	0.55	0.18	0.27	0.00	0.00	0.00
Sony_DSC_T77_0	0.00	0.00	0.00	0.00	0.00	0.00	0.78	0.08	0.13
Sony_DSC_T77_1	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.58	0.17
Sony_DSC_T77_2	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.06	0.56

used to pre-train the network. After several iterations, we load the pre-trained network to train all samples in Table.9 The accuracy curve is shown in Fig.8. From where we can see, the retrained process need fewer iterations to obtain the higher accuracy, which can greatly reduce the training time for new camera categories. Every 100 iterations take about 52 seconds when the batch size is set to 96.

VI. CONCLUSION

In this paper, we propose a new deep learning approach for solving the problem of category coupling and image attribute extracting. To accomplish such goal, image contents are extracted by the multiple convolution kernel. By subtracting the image contents from the original images, the images can be better classified. We adopt coupling coding method to train the network and the multi-classification problem is decomposed into a number of binary or tri-classification problems. Meanwhile, redundant coding can improve the scalability of the network. Pre-training experiments show that redundant coding has small effect on the classification performance. For several models which are difficult to classify, we proposed the auxiliary classifier and we believe that it can be better trained by appropriate setting crop position. We evaluate the effectiveness of our proposed methods by using the Dresden database. The final experiment shows that our method is superior to the existing methods.

REFERENCES

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[2] A. Piva, "An overview on image forensics," *ISRN Signal Process.*, vol. 2013, pp. 1–22, Jan. 2013.

[3] L. Gaborini, P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Multi-clue image tampering localization," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2014, pp. 125–130.

[4] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar, "Face swapping: Automatically replacing faces in photographs," *ACM Trans. Graph.*, vol. 27, no. 3, p. 39, 2008.

[5] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video face replacement," *ACM Trans. Graph.*, vol. 30, no. 6, p. 130, 2011.

[6] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3677–3685.

[7] G. Xu and Y. Q. Shi, "Camera model identification using local binary patterns," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 392–397.

[8] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "A study of co-occurrence based local features for camera model identification," *Multimedia Tools Appl.*, vol. 76, no. 4, pp. 4765–4781, Feb. 2017.

[9] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, "Automatic source camera identification using the intrinsic lens radial distortion," *Opt. Express*, vol. 14, no. 24, pp. 11551–11565, 2006.

[10] Z. Lin, J. He, X. Tang, and C.-K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognit.*, vol. 42, no. 11, pp. 2492–2501, Nov. 2009.

[11] B. Bayar and M. C. Stamm, "Towards open set camera model identification using a deep learning framework," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2007–2011.

[12] P. Yang, R. Ni, Y. Zhao, and W. Zhao, "Source camera identification based on content-adaptive fusion residual networks," *Pattern Recognit. Lett.*, vol. 119, pp. 195–204, Mar. 2019.

[13] Y. Chen, Y. Huang, and X. Ding, "Camera model identification with residual neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4337–4341.

[14] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[15] M. Jahanirad, A. W. A. Wahab, and N. B. Anuar, "An evolution of image source camera attribution approaches," *Forensic Sci. Int.*, vol. 262, pp. 242–275, May 2016.

[16] M. Al-Ani and F. Khelifi, "On the SPN estimation in image forensics: A systematic empirical evaluation," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 5, pp. 1067–1081, May 2017.

[17] H. Zeng and X. Kang, "Fast source camera identification using content adaptive guided image filter," *J. Forensic Sci.*, vol. 61, no. 2, pp. 520–526, Mar. 2016.

[18] S. Gao, G. Xu, and R.-M. Hu, "Camera model identification based on the characteristic of CFA and interpolation," in *Proc. Int. Workshop Digit. Watermarking*. Berlin, Germany: Springer, 2011, pp. 268–280.

[19] C. Riess and E. Angelopoulou, "Scene illumination as an indicator of image manipulation," in *Proc. Int. Workshop Inf. Hiding*. Berlin, Germany: Springer, 2010, pp. 66–80.

[20] M. G. Hwang, H. J. Park, and D. H. Har, "Source camera identification based on interpolation via lens distortion correction," *Austral. J. Forensic Sci.*, vol. 46, no. 1, pp. 98–110, Jan. 2014.

[21] J. Luka, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, Jun. 2006.

[22] Y. Li and C.-T. Li, "Decomposed photo response non-uniformity for digital forensic analysis," in *Proc. Int. Conf. Forensics Telecommun., Inf., Multimedia*. Berlin, Germany: Springer, 2009, pp. 166–172.

[23] C.-H. Choi, J.-H. Choi, and H.-K. Lee, "CFA pattern identification of digital cameras using intermediate value counting," in *Proc. 13th ACM Multimedia Workshop Multimedia Secur. (MM&Sec)*, 2011, pp. 21–26.

[24] B. Mahdian, R. Nedbal, and S. Saic, "Blind verification of digital image originality: A statistical approach," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1531–1540, Sep. 2013.

[25] F. Ahmed, F. Khelifi, A. Lawgalv, and A. Bouridane, "Comparative analysis of a deep convolutional neural network for source camera identification," in *Proc. IEEE 12th Int. Conf. Global Secur., Saf. Sustainability (ICGS)*, Jan. 2019, pp. 1–6.

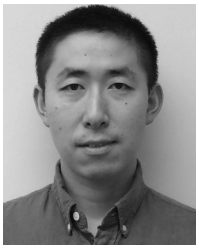
[26] V. U. Sameer, R. Naskar, N. Musthyala, and K. Kokkalla, "Deep learning based counter–forensic image classification for camera model identification," in *Proc. Int. Workshop Digit. Watermarking*. Cham, Switzerland: Springer, 2017, pp. 52–64.

[27] L. Bondi, L. Baroffio, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 259–263, Mar. 2017.

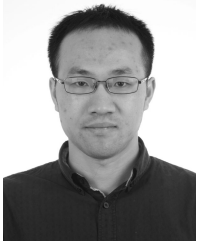
[28] D. Freire-Obregón, F. Narducci, S. Barra, and M. Castrillón-Santana, "Deep learning for source camera identification on mobile devices," *Pattern Recognit. Lett.*, vol. 126, pp. 86–91, Sep. 2019.
[29] G. B. Thomas, N. L. Eby, and J. M. Rago, "Using a mobile computing device camera to trigger state-based actions," U.S. Patent App. 10 068 221, Sep. 4, 2018.
[30] J. Spooren, D. Preuveneers, and W. Joosen, "Mobile device fingerprinting considered harmful for risk-based authentication," in *Proc. 8th Eur. Workshop Syst. Secur.*, 2015, p. 6.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[32] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
[33] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," in *Proc. ACM Symp. Appl. Comput.*, 2010, pp. 1584–1590.

MENGNAN ZHAO (Student Member, IEEE) received the B.S. degree in electronic and information engineering from the Tianjin University of Technology, China, in 2018. He is currently pursuing the master's degree with the School of Information and Communication Engineering, Dalian University of Technology. His research interests include adversarial samples and deep learning.

BO WANG (Member, IEEE) received the B.S. degree in electronic and information engineering and the M.S. and Ph.D. degrees in signal and information processing from the Dalian University of Technology, Dalian, China, in 2003, 2005, and 2010, respectively. From 2010 to 2012, he was a Postdoctoral Research Associate with the Faculty of Management and Economics, Dalian University of Technology, where he is currently an Associate Professor with the School of Information and Communication Engineering. His current research interests focus on the areas of multimedia processing and security, such as digital image processing and forensics.



FEI WEI (Member, IEEE) received the B.S. degree in electrical engineering from the Harbin University of Science and Technology, Harbin, China, in 2012, and the M.S. degree in electrical engineering from Harbin Engineering University, Harbin, in 2015. He is currently pursuing the Ph.D. degree in electrical engineering with The State University of New York at Buffalo, Buffalo, NY, USA. Since 2015, he has been a Research Assistant with the Department of Electrical Engineering, SUNY at Buffalo, Buffalo. His research interests include cross fields of network information theory, coding theory, machine learning, and data mining.



MEINENG ZHU (Member, IEEE) received the B.S. degree in telecommunication and the M.S. degree in pattern recognition and intelligent system from the Huazhong University of Science and Technology, Wuhan, China, in 2004 and 2007, respectively. He is currently an Associate Research Fellow with the Beijing Institute of Electronics Technology and Application. His current research interests focus on the areas of multimedia processing and security, such as digital image processing and forensics.

XUE SUI received the B.S. degree from Northeast Normal University, Changchun, China, in 1989, and the Ph.D. degree in education from Liaoning Normal University, Dalian, China, in 2004. From 2004 to 2006, he did postdoctoral research in basic medicine at Shantou University. He is currently a Professor with the College of Psychology, Liaoning Normal University. His areas of research are behavioral decision-making and neural mechanisms.