This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

Robust Multi-classifier for Camera Model Identification based on Convolution Neural Network

HONGWEI YAO[1], TONG QIAO[1,2], MING XU[1], AND NING ZHENG[1]
[1]Hangzhou Dianzi University, School of Cyberspace, Hangzhou, China
[2]Zhengzhou Science and Technology Institute, Zhengzhou, China

Corresponding author: Ming Xu (mxu@hdu.edu.cn).

This work is funded by the Cyberspace Security Major Program in National Key Research and Development Plan of China under grant No. 2016YFB0800201, the Natural Science Foundation of China under grant No. 61702150 and No. 61572165, the State Key Program of Zhejiang Province Natural Science Foundation of China under grant No. LZ15F020003, the Key Research and Development Plan Project of Zhejiang Province under grant No. 2017C01062 and No.2017C01065.

ABSTRACT With the prevalence of adopting data-driven Convolution Neural Network (CNN) based algorithms into the community of digital image forensics, some novel supervised classifiers have indeed increasingly spring up with nearly-perfect detection rate, compared to conventional supervised mechanism. The goal of this paper is to investigate a robust multi-classifier for dealing with one of image forensic problems, referring to as Source Camera Identification (SCI). The main contributions of this paper are threefold: (1) by mainly analyzing the image features characterizing different source camera models, we design an improved architecture of CNN for adaptively and automatically extracting characteristics, instead of hand-crafted extraction; (2) the proposed efficient CNN-based multi-classifier is capable of simultaneously classifying the tested images acquired by a large scale of different camera models, instead of utilizing a binary classifier; (3) numerical experiments show that our proposed multi-classifier can effectively classify different camera models with achieving an average accuracy nearly 100% relying on majority voting, which indeed outperforms some prior arts; meanwhile its robustness has been verified by considering that the images are attacked by post-processing such as JPEG compression and noise adding.

INDEX TERMS Camera model identification, deep learning, convolution neural network (CNN), passive image forensics

I. INTRODUCTION

With the development of digital technology, digital images can be conveniently acquired from various camera devices, and widely spread on social network platforms. Meanwhile, digital images can be easily manipulated using low-cost photo editing software, or the relative information linking between the image and digital camera can be maliciously removed by unlawful criminals or unauthorized organizations, resulting into the ownership infringement. Therefore, the study of designing reliable and robust forensic methods, which makes the community of digital image forensics receive an increasing attention, is urgently needed by the juridical organization.

A. STATE OF THE ART

In recent studies, passive (or blind) image forensics without requiring any embedded information such as digital watermark or signature, dominates the research community of digital image forensics. In general, passive image forensics is a technique mainly focusing on two following problems: image tampering authentication and image source identification (see a complete overview in [1]–[3]). The image tampering authentication mainly addresses the problem of detecting whether the image under investigation suffers attacks from image post-processing, such as re-sampling [4], splicing [5], copy-move forgery [6], median filtering [7], and JPEG compressing [8].

The problem of image source identification can also be defined as Source Camera Identification (SCI), which mainly investigates the origin of the images. Specifically, forensic investigators might devise the algorithms of SCI involving three significant details: camera brands, camera models, and even camera individual instances (see [9]–[11] for details). It should be noted that the problem of model identification is addressed by different camera models, possibly involving more than one instance for each given model. Still, we need

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

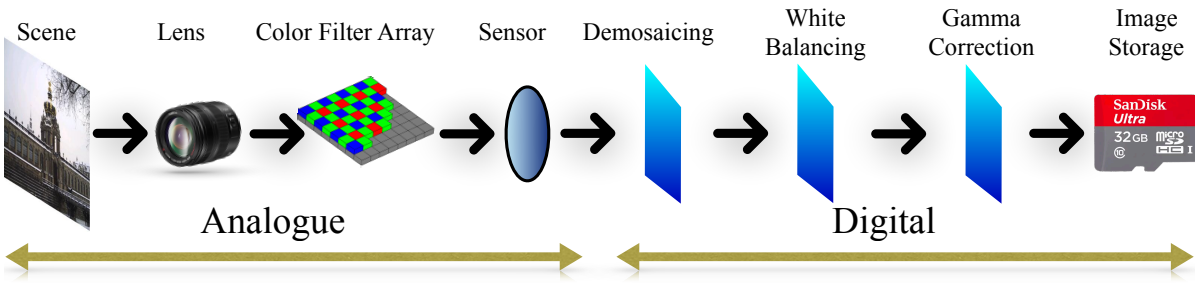

FIGURE 1: Illustration of a typical imaging pipeline within a digital still camera.

emphasize that the algorithm of individual instance identification focuses on distinguishing different camera instances with the same model. In this practical context, we propose to study the algorithm of camera model identification.

Generally, the captured digital images is stored within header files such as EXchangeable Image File (EXIF) and JPEG headers. All recording information is contained in the header files. Therefore, the forensic evidences can be easily accessed by extracting the concerning information from the header files, which serve as camera fingerprints. However, the header files can be feasibly removed or replaced by malicious criminals. Meanwhile, photographs posted on the social networks, do not carry the header files or any other information serving as camera fingerprints.

In that practical scenario, it is proposed to extract intrinsic fingerprints existing among stages of the image acquirement pipeline, involving the following primary steps: collecting the incident lights onto the lens, filtering color channels using a Color Filter Array (CFA) pattern, converting the incident lights into an electrical current using a sensor, and some other processing steps such as demosaicing, white balancing, gamma correction, etc. (see Fig. 1). The readers may refer to [12] for detail explanations. Then most of SCI algorithms have been deeply investigated in virtue of the image acquirement pipeline. In early studies, the estimation of the CFA pattern or demosaicing algorithm [13], [14], lens distortion [15], white balancing [16] have been utilized to design the discriminators. In most literature of current studies, the Sensor Pattern Noise (SPN) caused by the limitations of sensor manufacturing processes, mainly referring to as the inhomogeneity of silicon wafer (see [17], [18]), has always been proposed to design the general framework of an effective classifier.

Based on the SPN features, most forensic classifiers can be arbitrarily formulated into two categories: statistical model-based algorithm, often defined as un-supervised method; machine learning-based method, or named as supervised method. Then let us specifically extend those two categories of typical classifiers as follows.

- **Statistical model-based algorithm**: the authors of [19] first propose to utilize SPN, mainly referring to as Photo-Response Non-Uniformity (PRNU) noise, to design the classifier for identifying the source camera device. Afterwards, the up-dated version of that classifier with higher detection rate is established in [20]. Generally, the value of Peak to Correlation (PCE) directly serves as the threshold for discrimination. In fact, some other prior arts also concentrate their studies on improving the effectiveness and robustness of the PRNU-based framework. In addition, even though the authors [19], [20] have proposed to empirically evaluate the proposed model and analyze the performance of the detector, its theoretical performance is still unknown.
Till recently, the detectors of [21] propose to use a novel SPN-based noise extracted from RAW format images, in terms of Poisson-Gaussian-distributed noise, characterizing the features of each camera model, to establish the classification under the framework of hypothesis testing theory. Inspired by that work, the the detector of [22] has been proposed to address the problem of camera instance classification. Besides, the novel SPN-based classifiers can be extended into the design of classifying images based on JPEG format images [23], [24]. More importantly, the series of detectors [21]–[24] can theoretically give the upper bound of the detection at the prescribed false alarm probability, and theoretically established performance. Nevertheless, the statistical model-based algorithms only relying the unique feature (SPN) have the limitations: 1) binary classifiers cannot simultaneously discriminate different camera models; 2) noise model is heavily dependent of the image content, resulting into the unsatisfying robustness of the proposed classifier. In this context, our proposed robust multi-classifier with CNN can indeed overcome those limitations.
- **Machine learning-based algorithm**: algorithms in this category primarily rely on manually defined procedures with feature extraction. The data-oriented framework of learning-based algorithms not only extracts SPN, but also other SPN-related features for establishing the classifier. Generally, The labeled images with extracted features in the training stage are first used for training a discriminator; and then in the testing stage, the prior-trained classifier is used for identifying camera model or instance. In the stage of feature extraction,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

most of prior studies extracts a large scale of feature matrices manually, which is inefficient, resulting in unavoidable accuracy deviation. For instance, the limitation of Support Vector Machine (SVM) with associated learning algorithms is that setting of key parameters directly determines if the optimal classification results can be achieved [25]. Besides, the problem of robustness to mismatch between training and testing dataset remains open [19], [26]–[28]. The another limitation is that binary classification results in multiple operations when dealing with the problem of multi-classification (see [29]).

Compared to the SVM-based algorithms, CNN-based methods have unparalleled advantages in addressing the problem of feature extraction. The optimized CNN algorithm has the ability of modifying the typical weight of neurons through calculating the gradient of the designed loss function, which can learn feature representations automatically and effectively. In recent studies, due to the gradually improved superior capability of CNN dealing with the problem of classification, some forensic investigators propose a novel framework with using CNN algorithm (see [7], [30] for instance). To our knowledge, authors of [31] first propose to use CNN-based classifier of identifying camera model. Then the algorithm proposed in [32] further improves that pioneer work, and validates the practicability of adopting CNN to solve the problem of SCI. It should be noted that although the algorithm of [32] has achieved high accuracy, the designed classifier is still a binary classifier. Besides, owning to that the feature extraction is independent of classification process, the detection accuracy is affected and complexity of algorithm is increased. In this practical context, we propose to establish CNN-based multi-classifier with ability of simultaneously classifying different models. Note that by using less training data, our proposed algorithm slightly improves the detection rate of [32], and meanwhile the number of classification capability is up to 25 different models, larger than 18 models of [32].

B. CONTRIBUTIONS OF THE PAPER

In this paper, we investigate the problem of SCI and propose a novel CNN-based multi-classifier for camera model identification instead of utilizing a binary classifier. Different from prior supervised algorithms, we focus on improving the architecture of CNN by adopting a group of 3×3 and 1×1 kernel convolutional layers which extract feature maps adaptively and automatically. Finally, numerical experiments demonstrate that our algorithm not only outperforms some prior arts, but also performs its robustness when images are attacked by post-processing such as JPEG compression and noise adding.

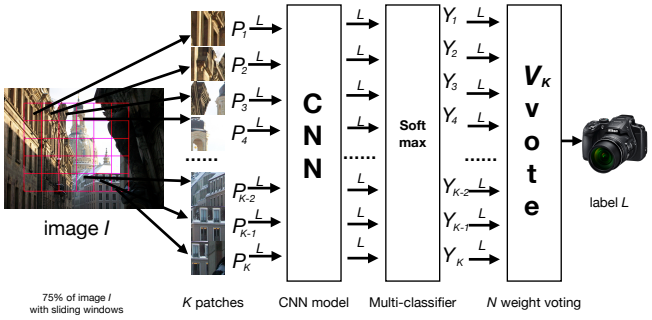

FIGURE 2: Illustration of training pipeline: from dividing $C\%$ of an image (the central portion) into K patches using sliding windows, extracting features through CNN and outputing feature vectors to multi-classifier, then exporting to N majority voting process.

C. ORGANIZATION OF THE PAPER

This paper is organized as follows. Sec. II gives the framework of our proposed CNN model involving patch selection, establishment of convolutional layer and classification layer. In Sec. III, we specifically describe our proposed CNN-based multi-classifier. Sec. IV presents the numerical results over the benchmark image dataset, and also demonstrates the comparison with prior arts. Besides, the robustness of our proposed algorithm is verified. Finally, Sec. V concludes this paper.

II. OUR PROPOSED CNN MODEL

In the following sections, we describe our model details (see Fig. 2): (1) patch selection, we split a full-size image into a set of non-overlapped patches and select high-quality patches from it; (2) convolutional layer, we discuss each component and structure of a convolutional layer; (3) classification layer, we investigate how classification layer uses the features extracted by convolutional layer, and then explain the architecture of our modified CNN. In general, (1) belongs to image pre-processing; (2) is used for extracting features; (3) serves for outputting prediction labels.

A. PATCH SELECTION

The first step of the proposed framework is to select patches from a three color-channel image I belonging to the camera model L, where L denotes one label of N given known camera models. It should be noted that each patch associated with the same label inherits from the same source image I. We put forward a "sliding window" algorithm by using a 64×64 square to crop central portion of the image I corresponding to K extraction of non-overlapping patches. Then let us denote each patch as $P_k, k \in \{1, \ldots, K\}$, that carries the relative labels L from the camera model L (see Fig. 2 for detailed illustration).

In this context, the proposed CNN model requires the setting of input patch size with $64 \times 64 \times 3$, in which the pixel intensity of each channel ranges from 0 to 255. Compared to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

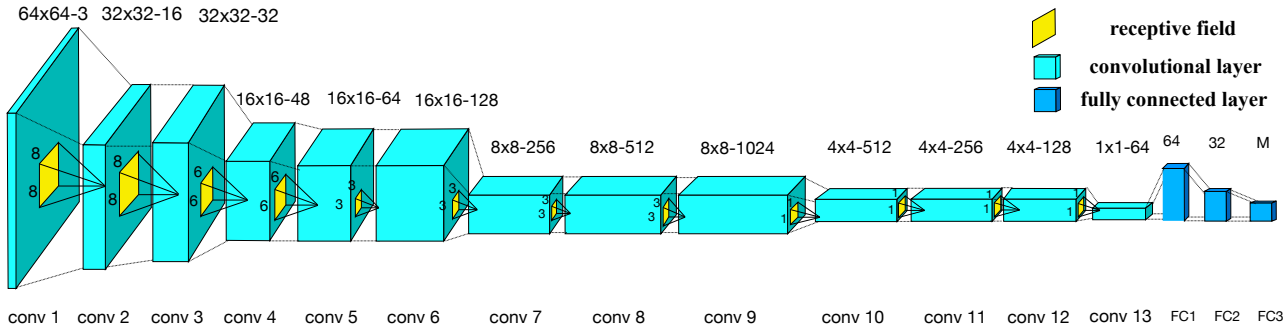

FIGURE 3: Architecture of the CNN, with 13 conventional layers, 3 fully connected layers, Table 1 describes detail configuration for every layers.

TABLE 1: Description details of CNN architecture layers configuration in Fig. 3.

ID	Input size	Configuration	Type
conv 1	64×64-3	stride=2, ksize=8×8	conv+ReLU
conv 2	32×32-16	stride=1, ksize=8×8	conv+ReLU
conv 3	32×32-32	stride=2, ksize=6×6	conv+ReLU
conv 4	16×16-48	stride=1, ksize=6×6	conv+ReLU+maxpool
conv 5	16×16-64	stride=1, ksize=3×3	conv+ReLU
conv 6	16×16-128	stride=2, ksize=3×3	conv+ReLU
conv 7	8×8-256	stride=1, ksize=3×3	conv+ReLU+maxpool
conv 8	8×8-512	stride=2, ksize=3×3	conv+ReLU
conv 9	8×8-1024	stride=2, ksize=3×3	conv+ReLU+maxpool
conv 10	4×4-512	stride=1, ksize=1×1	conv+ReLU
conv 11	4×4-256	stride=1, ksize=1×1	conv+ReLU
conv 12	4×4-128	stride=2, ksize=1×1	conv+ReLU
conv 13	1×1-64	stride=2, ksize=1×1	conv+ReLU+maxpool

current CNN model for SCI, the reason for using the "sliding window" algorithm can be explained as: (1) splitting images into numerous patches results into efficiently augmenting the number of training data; (2) feeding CNN model with patches (portions of an image), instead of a full-size image greatly reduces the size of CNN model, and meanwhile makes CNN model very sensitive to feature extraction. After the procedure of patch selection, we randomly shuffle all patches, which is used for feature extraction based on our proposed CNN model. In the next section, we mainly explore feature extraction containing many convolutional layers.

B. CONVOLUTIONAL LAYER

The establishment of a convolutional layer usually consists of two main stages: non-linear operation and linear convolution, where non-linear operation usually includes the design of activation function and pooling layer (see [7]). There are two theories critical for linear convolution: receptive field and shared weights. Receptive field refers to as the minimum size of convolution matrices for each iteration, on which the procedure of feature extraction mainly relies. The shared weights can cut down the number of parameters in the network, and improve the efficiency of the proposed network (see [33]). Besides, for the output of a convolutional layer, the activated domain of a convolutional layer is defined as feature map, and the shared weights of a convolution are used to define a filter.

A neural network without the activation function would be simplified to a linear regression model. It has less power to learn complex functional mappings from data, and does not perform well in the practical classification. Therefore, we add the activation function to each hidden layer in our proposed CNN model. Among numerous activation functions, rectification non-linearity (ReLU) [34]) has been verified to greatly accelerate the convergence of stochastic gradient descent (SGD) [35], and also performs very well in our designed multi-classifier. ReLU activates all output units with larger than zero, and meanwhile suppresses output units with smaller than zero, resulting into that it can convert computation-cost operations to simply limiting a matrix of activations to zero relying on the prescribed threshold. That property indeed helps our proposed multi-classifier improve the detection performance. In the next section, we will discuss the design of the classification layer with feature maps extracted by using convolutional layers.

C. CLASSIFICATION LAYER

In general, classification layer consisting of few fully connected layers is characterized by most of the network's parameters. However, the establishment of fully connected layers does not cost too much operation time. When convolutional layers extract the features, fully connected layer feeds feature maps back to Hierarchical Softmax (see [36]), that decomposes labels into a tree. Each label is then denoted as a path along the corresponding tree. Besides, it should be noted that a Softmax classifier trained at each node of the tree is in order to disambiguate between the left and right branch.

During the process of gradient descent, the back propagation algorithm constantly adjusts the model parameters to the top layer of the CNN model in virtue of tuning loss function. In that case, the CNN model can be trained to the optimal correctness. However, it is unavoidable that the phenomenon of overfitting occurs, which probably impairs detection performance of our proposed multi-classifier. To prevent that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

nuisance result, we denote a dropout regularization rate $p\%$, meaning that neurons with a probability of $(1 - p\%)$ are abandoned at each level for the first two fully connected layers. In the practical classification, the proposed dropout algorithm increases the training convergence time, but greatly improves the accuracy of our multi-classifier.

Last but not least, in our proposed architecture of CNN model, we try to consider the mechanism of the voting layer following Softmax (see Fig. 2), which helps our multi-classifier make a final judgement. When an inspected image is split into K patches, and to push into the trained CNN model, each patch can obtain a classification result denoted as a probability at the output layer. We formulate V_k majority voting for those K patches, and the voting result is defined as the final classified result using our multi-classifier. In the following section, let us specifically extend the design of the CNN-based multi-classifier for dealing with the problem of SCI.

III. ESTABLISHMENT OF SCI MULTI-CLASSIFIER

A typical CNN architecture consists of convolutional, pooling, and fully connected layers. When constructing a framework of CNN model, the network designer needs consider the problem of prescribing parameters, including the number of layers for each type, the order of layers, the other parameters of each layer (such as parameter initialization strategy, convolution reception fields size, input and output size of each layer). Fig. 3 illustrates the design of our proposed CNN architecture. In the following paragraphs, we mainly focus on strategies for designing the architecture, such as the effectiveness of depth and width for feature extraction, the design of receptive field for convolutional layer.

Nevertheless, the establishment of the CNN architecture is a delicate step, meaning that the basic principle of designing a network has to consider the characteristics of the input data. On the one hand, a sufficiently wide neural network with just a few hidden layers can approximately be formulated by any polynomial function assuming that enough training data is acquired (see [37] for instance). Wide network is good at memorization, leading to a strong capacity to remember more learned data. On the other hand, the benefit of multiple layer (deep) network is that they can learn features at various levels of abstraction and forecast information for the next level. For instance, when we train a deep CNN to classify images, edges will be integrated in the first several layers, and the next layers will automatically train themselves to identify the outline of an object in the image, then the next layers will learn even higher-order features such as the whole object. Since the input data, referring to as patches, are not very large, the neural network should be good at analyzing difference between the pixel and its neighboring counterparts, and have a strong predictive ability to characterize feature maps. In general, a too wide network architecture can not fully learn feature map; a too deep network architecture might cause increment of the computational complexity. Hence, our proposed network is neither too deep nor too wide.

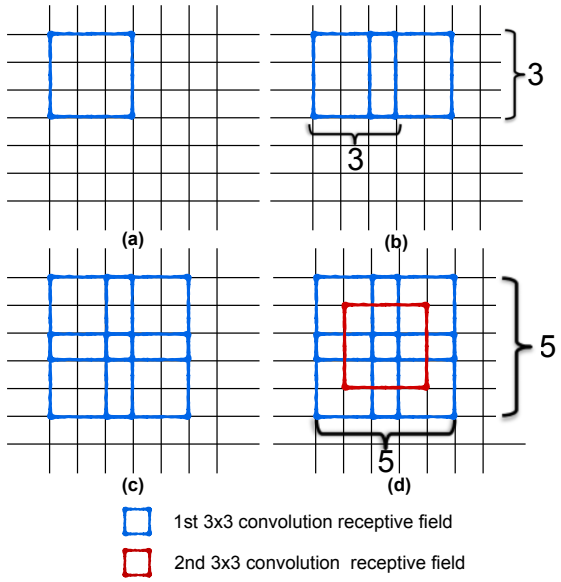

FIGURE 4: Stacked convolutional layers have a large receptive field. The example shows an image passing through two convolutional layers with 3×3 kernel size (or filter size) and applied with a stride of 2. After two 3×3 convolutional layers, the receptive field expands to 5×5. (a) Affected area at the first step of the first convolutional layer; (b) affected area of the second step; (c) affected area of the first convolutional layer; (d) affected area (blue region) and receptive field of the second convolutional layer (red region).

For the part of the receptive field, our CNN-based multi-classifier contains a group of convolutional layers with the kernel size of 3×3 plus 1×1, instead of a single 5×5 layer [32]. Fig. 4 gives the illustration by using this strategy. In the example, the effective area of the first convolutional layer becomes a 5×5 block in Fig. 4 (c) after the first iteration operation. In the second iteration operation, the block surrounded by the red line, which is inherited from the first convolutional layer (blue region) in Fig. 4 (c), is input of the second convolutional layer. Although the second convolutional layer kernel size is only 3×3, it has an effective area of 5×5. This example verifies that a stacked small kernel convolutional layers (3×3) can replace a single large kernel convolutional layer.

In addition, to furthermore describe the differences between the center pixel and the surrounding ones, we propose to add 1×1 convolution layers. It indeed increases the non-linearity of the decision function without interfering the receptive field of the convolutional layer. To our knowledge, Visual Geometry Group (VGG) [38] investigates the performance of 3×3 convolution filters, which shows that a significant improvement can be achieved by using 3×3 convolution filters, and deepen the neural network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

IV. NUMERICAL EXPERIMENTS

To demonstrate the effectiveness of our proposed method, we give numerical results: 1) we empirically demonstrate the performance of the proposed robust multi-classifier with CNN, and in comparison with current arts; 2) we evaluate the robustness of the proposed algorithm , with considering some practical attacks such as JPEG compression, noising adding, and image re-scaling.

To establish a comprehensive dataset for testing, we consider to use the Dresden image dataset [39], which are all JPEG format with quality factors (QF) over 75. That benchmark dataset is also widely-utilized by other forensic investigators for addressing the problem of SCI such as [40] and [32]. The dataset contains overall 27 camera models, and for some camera models, it contains multiple instances, consisting of a total of 74 camera instances. Table 2 and 3 respectively illustrate the camera statistic of the dataset. Due to limited data of Dresden image dataset, let us evaluate our proposed multi-classifier using 150 images for training and 150 images for testing camera model with multi-instances (see Table 2). Nevertheless, we believe that our proposed algorithm can still perform well over a large scale of dataset.

Afterwards, let us establish our experimental environment. All models run on a single Nvidia GPU card of type GeForce GTX 1070, with its built-in Deep Learning Tensorflow 1.4 module (see [41] for details).

TABLE 2: Experimental dataset serves for the results of Fig. 5. For each model, we randomly choose 150 images for testing dataset, 150 images for training dataset.

ID	Camera model	Resolution	Instance No.
1	Agfa DC-830i	3264×2448	1
2	Agfa Sensor530s	4032×3024	1
3	Canon Ixus70	3072×2304	2
4	Casio EX-Z150	3264×2448	3
5	FujiFilm FinePixJ50	3264×2448	3
6	Kodak M1063	3664×2748	3
7	Nikon CoolPixS710	4352×3264	3
8	Nikon D200	3872×2592	2
9	Nikon D70	3008×2000	2
10	Nikon D70s	3008×2000	2
11	Olympus mju 1050SW	3648×2736	3
12	Panasonic DMC-FZ50	3648×2736	3
13	Praktica DCZ5.9	2560×1920	3
14	Ricoh GX100	3648×2736	3
15	Rollei RCP-7325XS	3072×2304	3
16	Samsung L74wide	3072×2304	2
17	Samsung NV15	3648×2736	3
18	Sony DSC-H50	3456×2592	2
19	Sony DSC-T77	3648×2736	2
20	Sony DSC-W170	3648×2736	2

A. DETECTION PERFORMANCE OF CNN-BASED MULTI-CLASSIFIER

This section includes two experiments: the first experiment is designed to overall evaluate the detection performance of our proposed CNN-based multi-classifier for a single patch; the second one aims at validating the voting module performance of the multi-classifier. Meanwhile, we give the comparative

TABLE 3: Experimental dataset serves for the results of Fig. 6. For each model, we randomly choose 100 images for testing dataset, others for training dataset.

ID	Camera model	Resolution	Training No.	Testing No.
1	Agfa DC-504	4032×3024	132	100
2	Agfa DC-733s	3072×2304	160	100
3	Agfa DC-830i	3264×2448	156	100
4	Agfa Sensor505-x	2592×1944	131	100
5	Canon Ixus55	2592×1944	184	100
6	Canon Ixus70	3072×2304	152	100
7	Canon PowerShotA640	3648×2736	147	100
8	Casio EX-Z150	3264×2448	136	100
9	FujiFilm FinePixJ50	3264×2448	164	100
10	Kodak M1063	3664×2748	173	100
11	Nikon CoolPixS710	4352×3264	156	100
12	Nikon D200	3872×2592	175	100
13	Nikon D70	3008×2000	152	100
14	Nikon D70s	3008×2000	136	100
15	Olympus mju 1050SW	3648×2736	172	100
16	Panasonic DMC-FZ50	3648×2736	162	100
17	Pentax OptioW60	3648×2736	156	100
18	Praktica DCZ5.9	2560×1920	161	100
19	Ricoh GX100	3648×2736	152	100
20	Rollei RCP-7325XS	3072×2304	166	100
21	Samsung L74wide	3072×2304	145	100
22	Samsung NV15	3648×2736	154	100
23	Sony DSC-H50	3456×2592	164	100
24	Sony DSC-T77	3648×2736	137	100
25	Sony DSC-W170	3648×2736	164	100
$\sum$	-	-	3887	2500

results with other well-performed classifiers, such that from [40] and [32] (see Fig. 7). Because both of them are established based on machine leaning mechanism, and have achieved a high detection probability over 93%. In addition, it is proposed to analyze the limitations of two prior arts.

The algorithm of [40] extracts feature mainly relying on a rich model with describing a camera's demosaicing pattern. All the extracted features representing a labeled image acquired by a source camera model are trained for classification. In fact, the classifier of [40] has made remarkable results, however, there still exist some limitations: the overcomplex model used for feature extraction leads to relevant less efficient computation, especially dealing with a large full-size image. Besides, the problem of selecting effective hand-crated features remains open. To challenge that limitation, the algorithm of [32] uses CNN model to extract image features, and adopts a binary classifier based on SVM for solving the problem of SCI. Since the SVM is used to design a binary classification, computational complexity of classifiers with multi-SVM increases to $O\left(n^2\right)$. Furthermore, due to the separation between the feature extractor and the binary classifier, the system needs a large number of images for training the model.

We randomly select 25 camera models from Dresden image dataset, including more than 6000 images which are all natural JPEG compressed images (see Table 3). It should be noted the 18 camera models contain at least two instances (see Table 2). In particular, all images acquired by each camera model are split into two datasets, of which 100 for testing and the rest for training. Compared to the used number of training data from the method of [32], our proposed classifier uses less than two-third of the training data. Still,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

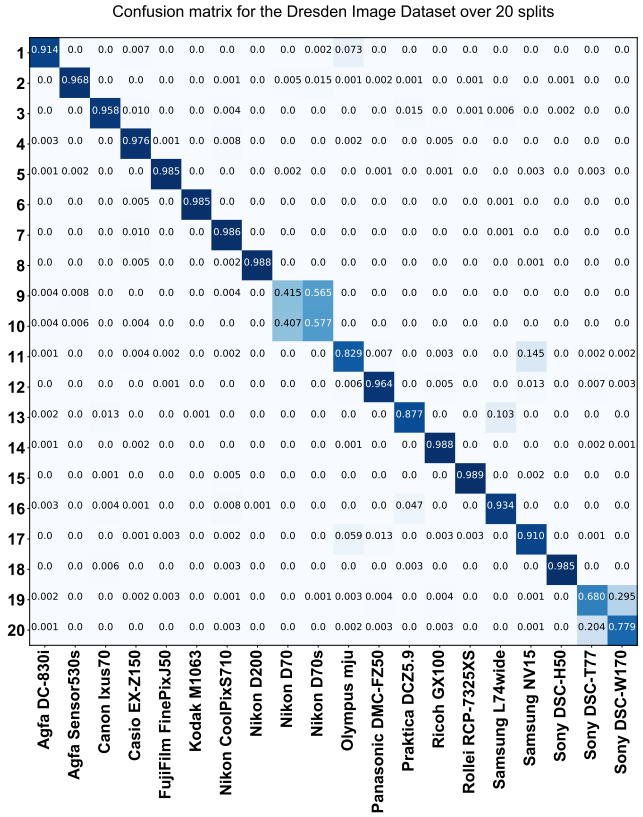

FIGURE 5: Confusion matrix for 20 camera models from Table 2 using the proposed method. Results are obtained with a single patch per image. Each cell reports the percentage of images from target class assigned to output class.

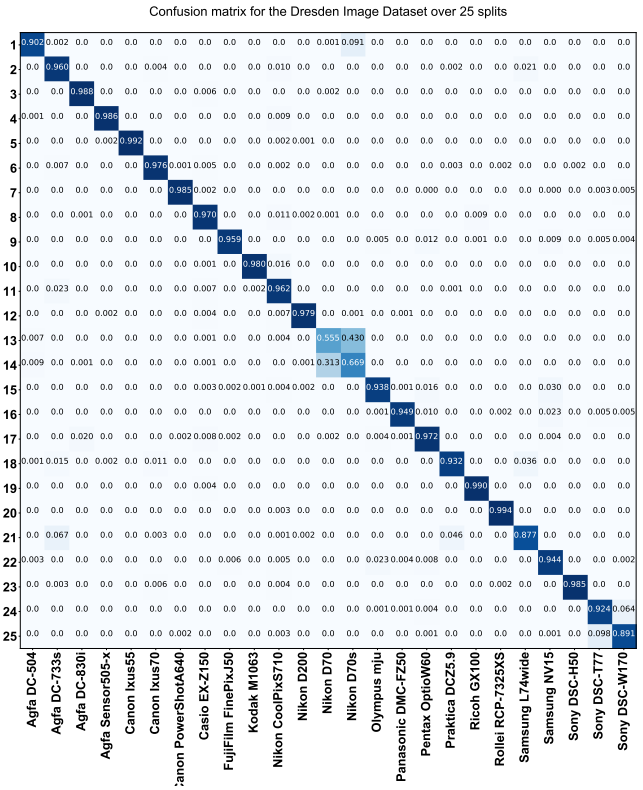

FIGURE 6: Confusion matrix for 25 camera models from Table 3 using the proposed method. Results are obtained with a single patch per image. Each cell reports the percentage of images from target class assigned to output class.

our algorithm demonstrates its efficient relevance with using less training samples.

In the procedure of training, we set $C = 75$ (see Fig. 2), referring to cropped patches from 75% (the central portion) of each image. Then $K = 256$ non-overlapping patches are extracted, where each patch size is 64×64 and containing nearly one million patches in total (3887×256. see Table 3).

Firstly, we evaluate the accuracy of the proposed method in classifying image patches (without voting). Fig. 5 illustrates the confusion matrix (consisting of probabilities) of detection perform using our proposed multi-classifier. In a confusion matrix, each row of the matrix represents the cameral model in a predicted label while each column (the output label) represents the one in an correct class. It should be noted that the correct rate (referring to as the probabilities along the main diagonal) is defined as the the predicted label is correctly classified as the output label. By observation, the average of the correct rate of our proposed multi-classifier is over 90%, which is slightly higher than that proposed by [40] or [32]. Besides, it should be noted that our proposed multi-classifier cannot distinguish between Nikon D70 and D70s. In fact, the settings of the twinborn models are very similar. By referring to [39], two identical lenses are used for the two digital cameras, Nikon D70 and D70s. Moreover, lenses can be interchanged for each acquired image between two (out of four) camera bodies. Thus, it is very difficult for us to distinguish them. However, we still try to deal with that problem in an alternative way. Let us classify the images acquired by D70 or D70s as the same source camera model, named as D70_70s in the first classification. Then, in the second classification, it is proposed to use the binary classifier of [24], which has the ability of distinguishing between Nikon D70 and D70s.

Furthermore, it is proposed to verify the effectiveness of our multi-classifier in the larger dataset with 25 camera models in Table 3. In this case, each model has one instance for testing. As Fig. 6 illustrates, the proposed multi-classifier remain its correct rate deal with distinguishing different camera models. It should be noted that the model DSC-T77 and DSC-W170 can be effectively discriminated while our designed classifier with 20 models (see Fig. 6) and the classifier of [32] cannot do that. More importantly, our modified CNN model has a strong ability to characterize camera models in a space with reduced dimensionality.

After validating the good performance on a single patch, we focus on the evaluation of the entire classification (with majority voting) in comparison with state-of-the-art methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

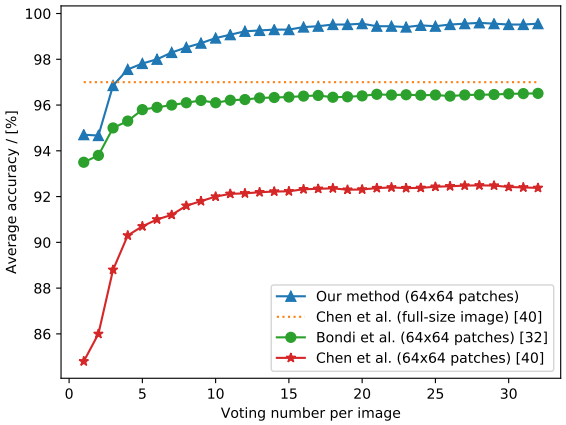

FIGURE 7: Accuracy comparison by varying the number of patches voting for each image of 20 camera models from Table 2.

Fig. 7 shows the average classification accuracy for dataset in Table 2, where the average accuracy is defined as:

$$Accuracy_v = \frac{1}{N} \times \sum_{l=1}^{N} y_{l,v}$$

where N denotes the number of given known camera models, $y_{l,v}$ is average accuracy for label l with the number of patches v, $v \in \{1, \ldots, 32\}$. As Fig. 7 illustrates, Our proposed multi-classifier is depicted by the blue line whose average accuracy gradually converges after the number of patches $v = 10$. The average accuracy is gradually enhanced (nearly close to 100% when $v = 32$) with the increment of voting number. Meanwhile, in comparison with the algorithms of Chen et al. [40] and of Bondi et al. [32], it's obvious that our designed multi-classifier can achieve better accuracy. Both two experimental results highlight that our algorithm slightly outperforms current arts, not only on the accuracy but also on the complexity of the algorithm.

B. ROBUSTNESS OF OUR PROPOSED METHOD

Since JPEG format has been widely adopted on social network platforms where the fingerprints of the images can be probably contaminated, it demands on the research of robustness of SCI algorithm.

In order to verify the robustness of our proposed algorithm, we still select images from Dresden dataset. Table 4 lists the static of used camera models that undergo unpredictable changes caused by content-preserving manipulations or geometric distortions, such as JEPG compression, adding noise and re-scaling. During the training process, we use the training dataset from Table 4 to train our CNN model without suffering any attacks. Then for the testing process, tested images are first pre-processed using attacks as: JPEG compression (case 1), or adding noise (case 2), or re-scaling (case 3), and then classified by our algorithm. In the following paragraphs, we will evaluate the robustness of our proposed multi-classifier by considering three cases.

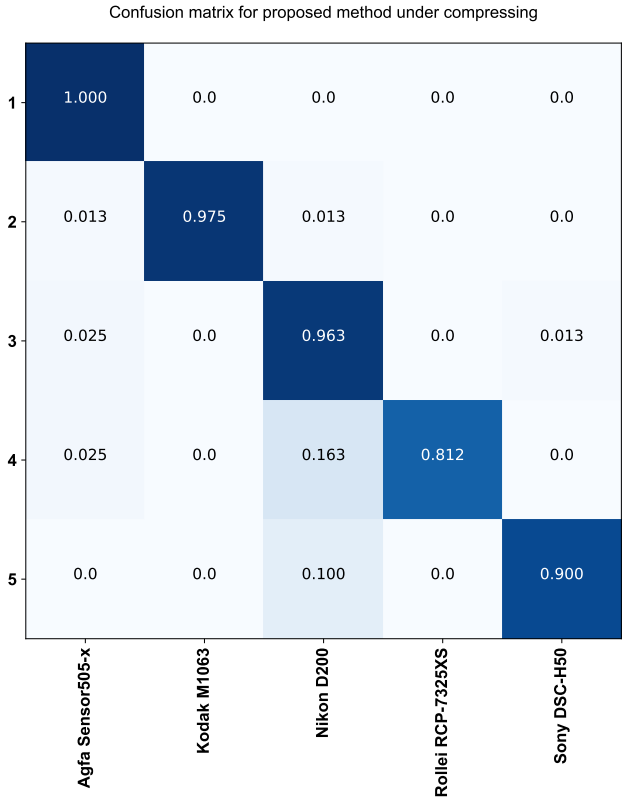

FIGURE 8: Confusion matrix for our proposed multi-classifier under the attack of JPEG compression with QF = 90.

TABLE 4: Experimental datasets. For each model, we randomly choose 80 images for testing dataset, others for training dataset, one device for each model.

Camera model	Resolution	Training No.	Testing No.
Agfa Sensor505-x	2592 × 1944	92	80
Kodak M1063	3664 × 2748	122	80
Nikon D200	3872 × 2592	118	80
Rollei RCP-7325XS	3072 × 2304	120	80
Sony DSC-H50	3456 × 2592	120	80

For case 1, let us evaluate the robustness of the multi-classifier under JPEG compression attack with different QFs. In the experiment, it is proposed to twice compress testing images with QF, ranging from 70 to 90 by steps of 10 (see Table 5). With decreasing QF, our multi-classifier remains its high detection accuracy. Even when the QF is 80, the average accuracy is still over 82%. In addition, Fig. 8 illustrates the confusion matrix with JPEG compression with QF = 90.

For case 2, it is proposed to add Gaussian-distributed random noise to tested images with different intensities, which are: $\mathcal{N}_1(0,1)$, $\mathcal{N}_2(0,2)$, $\mathcal{N}_3(1,1)$, $\mathcal{N}_4(1,2)$ (For $\mathcal{N}_i(x,y)$, where x denotes expectation, y denotes variance). From Table

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2832066, IEEE Access

TABLE 5: The results of the robustness for our multi-classifier under the attack by JPEG compression.

Camera model \ Quality factor	70	80	90
Agfa Sensor505-x	96.3	97.1	100
Kodak M1063	95.0	98.0	97.5
Nikon D200	98.8	94.8	96.3
Rollei RCP_7325X	58.8	70.5	81.2
Sony DSC-H50	62.5	78.4	90.0
Average accuracy	82.3	87.8	93.0

6, with increasing the intensity of adding noise, our multi-classifier sill performs very well. We can observe that the overall average accuracy reaches 86.22%, and the result directly verifies that our proposed method has the ability of achieving high accuracy, even with the interference of Gaussian-distributed noise.

TABLE 6: The results of the robustness for our multi-classifier under the attack by adding Gaussian-distributed noise.

Camera model \ Noise intensity	$\mathcal{N}_1(0,1)$	$\mathcal{N}_2(1,1)$	$\mathcal{N}_3(1,4)$	$\mathcal{N}_4(2,4)$
Agfa Sensor505-x	87.5	87.5	87.5	87.5
Kodak M1063	95.0	93.8	91.2	91.2
Nikon D200	88.7	91.2	86.3	81.2
Rollei RCP_7325X	90.0	90.0	90.0	95.0
Sony DSC-H50	86.3	82.5	83.8	76.2
Average accuracy	89.5	89.0	87.8	86.2

For case 3, we investigate the robustness of the proposed algorithm resisting against re-scaling attack. Table 7 demonstrates that in the case of re-scaling factor equal to 1.1, our multi-classifier can be effective when dealing with camera model Agfa Sensor505-x and Nikon D200, but fail in classification of other models.

TABLE 7: The results of the robustness for our proposed method under the attack of re-scaling.

Camera model \ Re-scaling factor	0.9	1.1
Agfa Sensor505-x	33.8	98.8
Kodak M1063	11.3	21.2
Nikon D200	23.7	88.7
Rollei RCP_7325X	8.8	8.8
Sony DSCH50	10.0	45.0
Average accuracy	17.5	52.5

In practice, up-loading users' images to website or transmitting images to each other both can result to some hidden attacks such as compression and re-scaling by servers of various social media platforms or noise adding by the information channel. To our knowledge, however, few state-of-the-art detectors consider the robustness dealing with the problem of SCI. In this context, the numerical experiments validate that our algorithm has power to resist against JPEG compression or adding noise. Unfortunately, in the case of re-scaling attack, our proposed multi-classifier can only be effective when dealing with two cameral models. However, it should be noted that few prior arts address the problem of considering the re-scaling attack.

V. CONCLUSIONS

In this paper we investigate a robust multi-classifier based on CNN model. Unlike current arts with binary classification algorithm, our proposed classifier can identify multiple camera models in one comparison, with high detection accuracy and strong robustness. Specifically, we focus on strategies for pre-processing images (patch selection), and designing a neural network architecture. Numerical experiments show that our proposed method can classify camera model accurately with one camera instance (see Fig. 6). However, when dealing with the case of Nikon D70/D70s, our multi-classifier cannot perform very well. Meanwhile, for the dataset (see Table 2) with more than one camera instance, our detector can effectively classify camera model except Sony DSC-T77/DSC-W170 and Nikon D70/D70s (see Fig. 5), and achieve an average accuracy nearly 100% when considering majority voting. Robustness experiments validate that our algorithm can resist against JPEG compression or adding noise. Unfortunately, it might be invalid when suffering re-scaling attacks. In future work, we will extend our algorithm to user identity identification using images shared on social network.

REFERENCES

[1] H. T. Sencar and N. Memon, Digital image forensics: There is more to a picture than meets the eye. Springer, 2012.
[2] M. C. Stamm, M. Wu, and K. Liu, "Information forensics: An overview of the first decade," IEEE Access, vol. 1, no. 1, pp. 167–200, 2013.
[3] P. Korus, "Digital image integrity–a survey of protection and verification techniques," Digital Signal Processing, vol. 71, pp. 1–26, 2017.
[4] T. Qiao, A. Zhu, and F. Retraint, "Exposing image resampling forgery by using linear parametric model," Multimedia Tools and Applications, vol. 77, pp. 1501–1523, 2017.
[5] H. Yao, S. Wang, X. Zhang, C. Qin, and J. Wang, "Detecting image splicing based on noise level inconsistency," Multimedia Tools and Applications, vol. 76, no. 10, pp. 12 457–12 479, 2017.
[6] X. Bi and C.-M. Pun, "Fast reflective offset-guided searching method for copy-move forgery detection," Information Sciences, vol. 418, pp. 531–545, 2017.
[7] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," IEEE Signal Processing Letters, vol. 22, no. 11, pp. 1849–1853, 2015.
[8] T. H. Thai, R. Cogranne, F. Retraint et al., "Jpeg quantization step estimation and its applications to digital image forensics," IEEE Transactions on Information Forensics and Security, vol. 12, no. 1, pp. 123–133, 2017.
[9] A. Lawgaly and F. Khelifi, "Sensor pattern noise estimation based on improved locally adaptive dct filtering and weighted averaging for source camera identification and verification," IEEE Transactions on Information Forensics and Security, vol. 12, no. 2, pp. 392–404, 2017.
[10] R. Li, C.-T. Li, and Y. Guan, "Inference of a compact representation of sensor fingerprint for source camera identification," Pattern Recognition, vol. 74, pp. 556–567, 2018.
[11] H. Zeng, J. Liu, J. Yu, X. Kang, Y. Q. Shi, and Z. J. Wang, "A framework of camera source identification bayesian game," IEEE transactions on cybernetics, vol. 47, no. 7, pp. 1757–1768, 2017.
[12] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," IEEE Signal Processing Magazine, vol. 22, no. 1, pp. 34–43, 2005.
[13] H. Cao and A. C. Kot, "Accurate detection of demosaicing regularity for digital image forensics," IEEE Transactions on Information Forensics and Security, vol. 4, no. 4, pp. 899–910, 2009.

[14] A. Swaminathan, M. Wu, and K. R. Liu, "Nonintrusive component forensics of visual sensors using output images," IEEE Transactions on Information Forensics and Security, vol. 2, no. 1, pp. 91–106, 2007.

[15] K. San Choi, E. Y. Lam, and K. K. Wong, "Source camera identification using footprints from lens aberration," in Electronic Imaging 2006. International Society for Optics and Photonics, 2006, pp. 60 690J–60 690J.

[16] Z. Deng, A. Gijsenij, and J. Zhang, "Source camera identification using auto-white balance approximation," in Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011, pp. 57–64.

[17] J. Nakamura, Image sensors and signal processing for digital still cameras. CRC press, 2005.

[18] J. R. Janesick, Scientific charge-coupled devices. SPIE press, 2001, vol. 83.

[19] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," IEEE Transactions on Information Forensics and Security, vol. 1, no. 2, pp. 205–214, 2006.

[20] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2009, pp. 72 540I–72 540I.

[21] T. H. Thai, R. Cogranne, and F. Retraint, "Camera model identification based on the heteroscedastic noise model," Image Processing, IEEE Transactions on, vol. 23, no. 1, pp. 250–263, 2014.

[22] T. Qiao, F. Retraint, R. Cogranne, and T. H. Thai, "Source camera device identification based on raw images," in Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015, pp. 3812–3816.

[23] T. H. Thai, F. Retraint, and R. Cogranne, "Camera model identification based on the generalized noise model in natural images," Digital Signal Processing, vol. 48, pp. 285–297, 2016.

[24] T. Qiao, F. Retraint, R. Cogranne, and T. H. Thai, "Individual camera device identification from jpeg images," Signal Processing: Image Communication, vol. 52, pp. 74–86, 2017.

[25] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," Journal of Machine Learning Research, vol. 11, no. Jul, pp. 2079–2107, 2010.

[26] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on. IEEE, 2008, pp. 1296–1299.

[27] X. Lin and C.-T. Li, "Preprocessing reference sensor pattern noise via spectrum equalization," IEEE Transactions on Information Forensics and Security, vol. 11, no. 1, pp. 126–140, 2016.

[28] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," IEEE Transactions on Information Forensics and Security, vol. 5, no. 2, pp. 280–287, 2010.

[29] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan, and M. javad Rajabi, "Advantage and drawback of support vector machine functionality," in Computer, Communications, and Control Technology (I4CT), 2014 International Conference on. IEEE, 2014, pp. 63–65.

[30] R. Beverly, S. Garfinkel, and G. Cardwell, "Forensic carving of network packets and associated data structures," digital investigation, vol. 8, pp. S78–S89, 2011.

[31] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in Information Forensics and Security (WIFS), 2016 IEEE International Workshop on. IEEE, 2016, pp. 1–6.

[32] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," IEEE Signal Processing Letters, vol. 24, no. 3, pp. 259–263, 2017.

[33] M. Browne and S. S. Ghidary, "Convolutional neural networks for image processing: an application in robot vision," in Australasian Joint Conference on Artificial Intelligence. Springer, 2003, pp. 641–652.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097–1105.

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in neural information processing systems, 2013, pp. 3111–3119.

[37] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," Neural networks, vol. 6, no. 6, pp. 861–867, 1993.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[39] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," Journal of Digital Forensic Practice, vol. 3, no. 2-4, pp. 150–159, 2010.

[40] C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features," in Information Forensics and Security (WIFS), 2015 IEEE International Workshop on. IEEE, 2015, pp. 1–6.

[41] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., "Tensorflow: A system for large-scale machine learning." in OSDI, vol. 16, 2016, pp. 265–283.