Camera Model Identification With The Use of Deep Convolutional Neural Networks

Amel TUAMA
LIRMM(UMR5506) /CNRS,
Montpellier University, FRANCE
Email: amel.tuama@lirmm.fr

Frédéric COMBY
LIRMM(UMR5506) /CNRS,
Montpellier University, FRANCE
Email: comby@lirmm.fr

Marc CHAUMONT
LIRMM(UMR5506) /CNRS,
Nîmes University, FRANCE
Email: marc.chaumont@lirmm.fr

***Abstract*—In this paper, we propose a camera model identification method based on deep convolutional neural networks (CNNs). Unlike traditional methods, CNNs can automatically and simultaneously extract features and learn to classify during the learning process. A layer of preprocessing is added to the CNN model, and consists of a high pass filter which is applied to the input image. Before feeding the CNN, we examined the CNN model with two types of residuals. The convolution and classification are then processed inside the network. The CNN outputs an identification score for each camera model. Experimental comparison with a classical two steps machine learning approach shows that the proposed method can achieve significant detection performance. The well known object recognition CNN models, AlexNet and GoogleNet, are also examined.**

***Index Terms*—Camera Identification, Deep Learning, Convolutional Neural Network, Fully Connected Network.**

I. Introduction

Source camera identification is the process of determining which camera device has been used to capture an image. It is used in security and legal issue as an evidence [1]. As a relation to prior work, researchers have proposed to use the artifacts that exist in the camera pipeline to collect specific features manually and use them to distinguish between camera models or individual devices.

We can classify the camera identification approaches in two families. The first family groups the methods that require to compute a model (PRNU, radial distortion) to identify a camera and then evaluate a statistical proximity (correlation) between the model and the image to test. Lukas et al. [2] propose a source camera device identification using the sensor pattern noise as a fingerprint for uniquely identifying sensors. Choi et al. [3] use the lens radial distortion. Since each camera model expresses a unique radial distortion pattern, it is used as a fingerprint to help on its identification. Dirik et al. [4] use the sensor dust patterns in digital single lens reflex cameras (DSLR) as a method for device identification.

The second family regroups the methods based on machine learning and feature vector extraction. Here, the model is built by the classification algorithm knowing the features. In order to identify a camera, the classifier evaluate the proximity (distance) between a previously learned model, and the feature vector of the image to test. Bayram et al. [5] determine the correlation structures presented in each color band in relation with the CFA interpolation. Kharrazi et al. [6] extract 34 features (color features, Image Quality Metrics (IQM), and wavelet domain statistics) and used them to perform camera model identification. Celiktutan et al. [7] use a subset of Kharrazi's feature sets and the features of binary similarity measures to identify the source cell-phone camera. Filler et al. [8] introduce a method of camera model identification from features of the statistical moments and correlations of the linear PRNU pattern. Gloe et al. [9] used Kharrazi's feature sets with extended color features to identify camera models. Xu and Shi [10] also proposed the camera identification using machine learning through Local Binary Patterns as features. Wahab et al. [11] used the conditional probability as a single feature set to classify camera models. Marra et al. [12] proposed 338 SPAM features from the Rich Models [13] based on co-occurrences matrices of image residuals. Tuama et al. [14] developed a method for digital camera model identification by extracting three sets of features: co-occurrences matrix, traces of color dependencies features related to CFA interpolation arrangement, and conditional probability statistics.

From the state of the art mentioned above, CNN approach has not been used for camera identification. In the field of digital forensics Bayar et al. [15] proposed a deep learning approach to detect image manipulation, while Chen et al. [16] introduced the convolutional neural networks in median filtering forensics. The general focus of machine learning is the representation of the input data and the generalization of the learning patterns. Good data representation can lead to high performance. Thus the key point is to construct features and data representations from raw data. Feature design consumes a large portion of the effort in a machine learning task, and is typically domain specific.

Deep Learning algorithms are one of the promising research fields into the automated extraction of complex data representations at high levels of abstraction. A key benefit of deep learning is that the analysis and learning of massive amounts of unsupervised data make it a valuable tool for Big Data Analysis. Thus, deep learning often produces good results [17]. Nevertheless, we must say that deep learning approaches require high computing resources compared to more traditional machine learning approaches. Indeed it necessitates a powerful GPU and a big database.

However, using a CNN as a black box leads to a weak performance in identifying camera model. Thus in this paper,

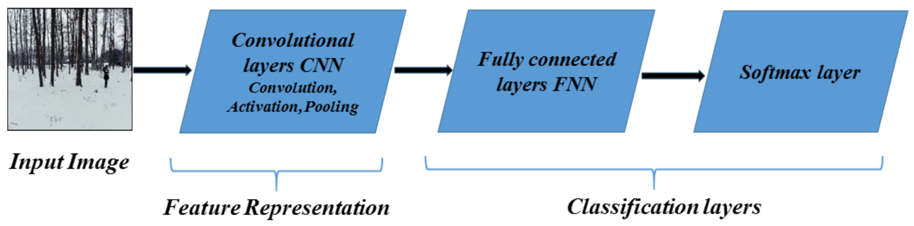

Fig. 1. The Conventional Neural Networks Concept.

we evaluate the obtained gain to modify the CNN model proposed by Krizhevsky [18]. We also experimentally compare our CNN model to AlexNet [18], and to GoogleNet [19].

The rest of this paper is organized as follows. Section II explains the concept of CNN and its relation to general machine learning concept. Section III presents all the details of our best CNN architecture for camera model identification. While in Section IV, we describe the experiments and the results. Conclusion comes in Section V.

II. Convolutional Neural Networks CNNs

Recently, Deep learning with the use of Convolutional Neural Networks (CNNs) have achieved wide interest in many fields. Deep learning frameworks are able to learn feature representations and perform classification automatically from original images. Convolutional Neural Networks (CNNs) have shown impressive performances in artificial intelligence tasks such as object recognition and natural language processing [20].

The general structure of a CNN consists of layers composed of neurons. A neuron takes input values, does computations and passes the result to the next layer. The general structure of a CNN is illustrated in Figure 1 which also shows the similarities with traditional machine learning approach. The next subsections describe the CNNs layers.

A. Convolutional layers & Classification layers

A conventional layer consists of three operations: *convolution*, the *activation function*, and *pooling*. The result of a convolutional layer is called a feature map, and can be considered as a particular feature representation of the input image. The *convolution* can be formulated as follows:

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

where $*$ denotes convolution, a_j^l is the j-th output map in layer l, w_{ij}^{l-1} is convolutional kernel connecting the i-th output map in layer $l-1$ and the j-th output map in layer l, b_j^l is the training bias parameter for the j-th output map in layer l, and n is the number of feature maps from layer $l-1$.

The *activation function* is applied to each value of the filtered image. There are several types of the activation function such as, an absolute function $f(x) = |x|$, a sine function $f(x) = sinus(x)$, or Rectified Linear Units (ReLU) function $f(x) = max(0, x)$.

The next important step is the *pooling*. A pooling layer is commonly inserted between two successive convolutional layers. Its function is to reduce the spatial size of the representation and to reduce the amount of parameters and computation in the network. During the pooling, a maximum or an average is computed.

The last process done by a convolutional layer is the normalization of the feature maps. The normalization is applied on the feature maps in order to obtain comparable output values for each neuron.

The classification layer consists of fully connected layers and a softmax function. In a fully connected layer, neurons have full connections to all activations in the previous layer. The activations can be computed with a matrix multiplication followed by a bias offset. The fully connected layer will compute the class scores by the softmax function. In this way, CNNs transform the original image from pixel values to the final class scores [17].

B. Learning process

When the learned features pass through the fully connected layers, they will be fed to the top layer of the CNNs, where a softmax activation function is used for classification. The back propagation algorithm is used to train the CNN. The weights and the bias can be modified in the convolutional and fully connected layers due to the error propagation process. In this way, the classification result can be fed back to guide the feature extraction automatically and the learning mechanism can be established.

The CNN architecture has millions of parameters which may arise overfitting problem. Drop out technique is used for reducing overfitting. It consists of setting the output of each hidden neuron with probability 0.5 to zero. The neurons which are dropped out in this way do not contribute to the forward pass and do not participate in backpropagation. This technique increases robustness, since a neuron can not rely on the presence of particular other neurons. It is, therefore, forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons [20].

III. The Proposed CNN Design for Camera Model Identification

The framework of our proposed model is shown in Figure 2, where we describe the detailed settings of the architecture. The

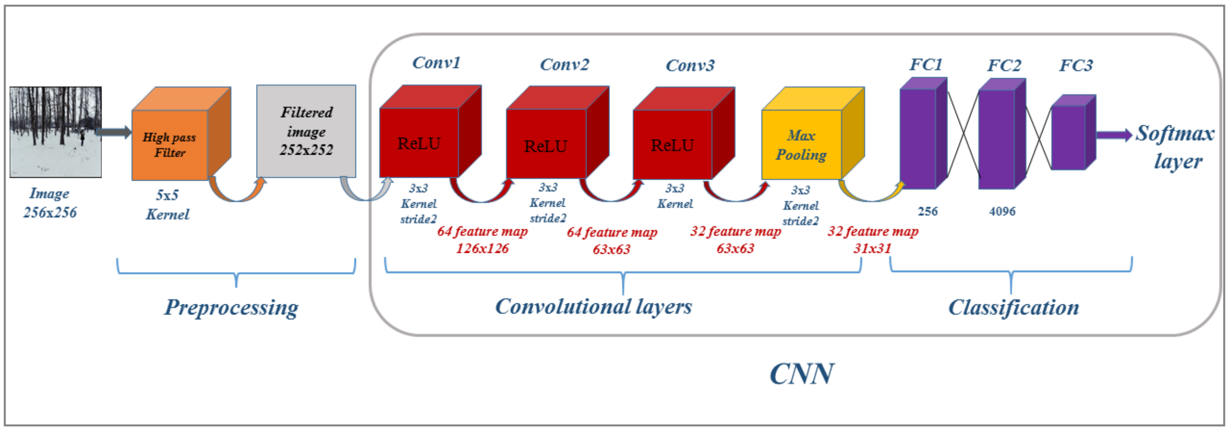

Fig. 2. The layout of our Conventional Neural Networks for Camera Model Identification.

first layer is the filter layer, followed by three convolutional layers from the first (Conv1) to the third (Conv3). While the last three layers are the fully-connected layers (FC1, FC2, FC3) for the classification. The details of our CNN model is illustrated in the following subsections.

A. Filter layer

The classical way for denoising an image is to apply a denoising filter. For each image I, the residual noise is extracted by subtracting the denoised version of the image from the image itself as follows:

$$N = I - F(I), \tag{2}$$

where $F(I)$ is the denoised image, and F is a denoising filter. This filter will be used in our experiments and applied on each color channel separately.

Another denoising high-pass filter is used on the input image I. This filter is the one used by Qian et al [21]. Applying this type of filter is important in the proposed method since it can suppress the interference caused by image edges and textures in order to obtain the image residual as follows:

$$A = I * \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \tag{3}$$

The output of this step will fed the CNN. In our experiments, we examined two types of filters as a preprocessing. The first one is the high pass filter adopted by Qian et al [21] and the second one is the well known wavelet based denoising filter [22].

B. Convolutions

AlexNet Convolutional Neural Networks [18] is adapted and modified to fit the model requirements. The first convolutional layer (Conv1) treats the residual image with 64 kernels of size 3×3. The size the feature maps produced is 126×126. Then the second convolutional layer (Conv2) takes the output of the first layer as the input. It applies convolutions with kernels of size 3×3 and produces feature maps of size 64×64. The third convolution layer applies convolutions with 32 kernels of size 3×3. The Rectified Linear Units (ReLUs) is a non-linearity activation function which is applied to the output of every convolutional layer. ReLUs is considered as the standard way to model a neurons output and it can lead to fast convergence with large models trained on large datasets [18].

The third convolutional layer is followed by a max pooling operation with window size 3×3 , which operate on the feature map in the corresponding convolutional layer, and lead to the same number of feature map with decreasing spatial resolution.

C. Fully Connected layers

The fully-connected layers (FC1) and (FC2) have 256, and 4096 neurons respectively. ReLUs activation function is applied to the output of fully connected layer. Each of (FC1) and (FC2) are dropped out during the learning. The output of last fully connected layer (FC3) is fed to a softmax function.

IV. Experiments and Evaluation

For the evaluation of the experiments, we used 33 camera models from two different data sets. The first set is made of 27 camera models from Dresden database [23], and the second set is 6 personal camera models. The list is given in Table I with the notice that all the images of the same model came from the same device. Using such different data sets ensure the diversity in the used data base. Before any further manipulation, The data set is subdivided into training and testing sets, such that 80% of the data set is chosen for the training and the rest 20% for the testing data.

In order to fit the CNN model conditions, we sub-divided the chosen data set images into 256×256 and we ignored those of less than 256×256. By applying the images sub-division step, we obtain a bigger data set which is beneficial for the training process. When doing the training/testing subdivision into two sets, we make sure that different parts of the same

TABLE I
CAMERA MODELS USED IN THE EXPERIMENTS, MODELS MARKED WITH * ARE OF PERSONAL CAMERA MODELS WHILE ALL THE OTHERS ARE FROM DRESDEN DATABASE.

Seq.	Brand	Model	Original Resolution	No. images 256×256
1	Agfa Photo	DC-733s	3072x2304	30349
2	Agfa Photo	DC-830i	3264x2448	39204
3	Agfa Photo	Sensor 530s	4032x3024	55585
4	Canon	Ixus 55	2592x1944	15680
5	Fujifilm	FinePix J50	3264x2448	22680
6	Kodak	M1063	3664x2748	64960
7	Nikon	D200 Lens A/B	3872x2592	55800
8	Olympus	M1050SW	3648x2736	28560
9	Panasonic	DMC-FZ50	3648x2736	37100
10	Praktica	DCZ 5.9	2560x1920	14630
11	Samsung	L74wide	3072x2304	24948
12	Samsung	NV15	3648x2736	30380
13	Sony	DSC-H50	3456x2592	36920
14	Sony	DSC-W170	3648x2736	28700
15	Agfa Photo	DC-504	4032x3024	10074
16	Agfa Photo	Sensor505-x	2592x1944	12040
*17	Canon	EOS-1200D	3648x2736	26780
*18	Canon	PowerShot SD790 IS	3648x2736	30016
19	Canon	Ixus70	3072x2304	20196
20	Canon	PowerShotA640	3648x2736	26320
*21	Canon	EOS7D	3648x2736	9360
22	Casio	EX-Z150	3264x2448	19548
23	Nikon	CoolPixS710	4352x3264	37944
24	Nikon	D70	3008x2000	13860
25	Nikon	D70s	3008x2000	13706
*26	Nikon	D5200	3648x2736	34500
27	Pentax	OptioA40	4000x3000	27885
28	Pentax	OptioW60	3648x2736	26880
29	Ricoh	GX100	3648x2736	26880
30	Rolli	RCP-7325XS	3072x2304	21384
*31	Sony	DSC-HX50	3648x2736	15960
*32	Sony	DSCHX60V	3648x2736	44400
33	Sony	T77	3648x2736	25340

original image do not belong, in the same time, to the training and testing sets. Table I shows all camera models with their number of images. For each experiment, the data set is chosen randomly and the results are averaged after running the procedure 5 times with 5 different splitting of the database.

The experiments are done with a single GPU card of type GeForce GTX Titan X manufactured by Nvidia, and DIGITS training system. Many experiments were done to achieve the design of the CNN model. We measure the efficiency of the CNNs by looking at the minimum error rate after convergence. Our CNN model is shown in Figure 2 and detailed in Section III. By applying two different filters, explained in subsection III-A, we have two different residuals which are referred to as $Residual1$ (high-pass filter), and $Residual2$ (wavelet noise filter) in the three different experiments.

Experiment 1

The first experiment uses the first 12 camera models given in Table I. For each image in the data set, a $residual1$ is extracted by applying a high pass filter [21]. Our CNN model is trained on the resulted residuals of the 12 camera models.

TABLE II
RESULTS FOR THE FIRST 12 CAMERA MODELS CONSIDERING THE POOLING PROCESS FOR $Residual1$.

Proposed Method	Accuracy
Two convolutional layer without Pooling	93.88%
Two convolutional layer with max Pooling	94.23%
Three convolutional layer with max Pooling	98.0%

Then we use it to identify the source camera model of each image in the test set to construct the identification accuracy. The confusion matrix of the classification results are shown in Table III. The average accuracy achieved by this experiment is 98%. From Table III, we can see that the best identification accuracy is recorded for the camera model $Kodak - M1063$ which achieves 99.89%. $Agfa - Sensor - 530s$, $Canon - 55$, $Fujifilm - FinePix - J50$, $Panasonic - DMC - FZ50$, and $Samsung - L74wide$ also achieved semi-perfect accuracy rates. While $Praktica - DCZ5.9$ recorded the least accuracy rate which is 90.44%.

Before going on in the experiments, it is important to evaluate the influence of the pooling layer. By adding a pooling layer for two convolutional layers, we achieve 94.23%, whereas without pooling, it was 93.88% . This result increased to 98.09% for three convolutional layers with max-pooling layer. The results of adding a pooling layer to the model is resumed in Table II.

The experiments reference as $Residual2$ is obtained by applying a wavelet denoising filter [2] on each image in the data set, then subtract the denoised image from the original one. Residuals of the training set fed the CNN model to perform the training process. This part achieves 95.1% as total identification accuracy for the 12 camera models which is 3% lower compared to $Residual1$. The confusion matrix of this part are shown in Table IV. With $Residual2$, the best identification accuracy is recorded for the camera model $Panasonic - DMC - FZ50$ which achieves 99.46%. While $Praktica - DCZ5.9$ recorded the least accuracy rate which it is 81.54%. We can hypothesize that the residuals obtained from such a filter suppress too much features related to some characteristic of the acquisition pipeline of a given camera model like the CFA interpolation, or lens-aberration correction traces, and that is exactly what the CNN model need to learn about the camera model features.

Experiment 2

The experiment is re-performed on the first 14 camera models of Table I, by adding $SonyDSC - H50$ and $SonyDSC - W170$ to the data set of experiment 1. This experiment achieved 97.09%, and 93.23% as a total identification accuracy for $residual1$, and $residual2$ respectively. The total identification accuracy is shown in Table V. The identification accuracy decreased with these two models due to the fact that the captured images from camera models of the same manufacturer are sometimes harder to separate, such as $SonyDSC - H50$ and $SonyDSC - W170$. This is due, as it has been observed in [1], to the strong feature similarity of some camera models from the same manufacturer.

Experiment 3

The proposed CNN model is performed again with all the 33 camera models given in Table I. We achieve 91.9% as an identification accuracy for the 33 camera models for $Residual1$. As we can see, the accuracy is decreased as the number of models is increased, and this is a known behavior in machine learning approach, especially when increasing the number of classes [9]. For $Residual2$, the experiments are less useful since the results are lower compared to $Residual1$. The results for the three data sets of camera models (12,14,33) are shown in Table V.

Comparison with AlexNet and GoogleNet

AlexNet was developed by Alex Krizhevsky et al. [18], and GoogleNet was designed by Szegedy et al. [19]. These two CNNs models are trained on our data sets to be compared with our proposed CNN model. The results are illustrated in Table V. GoogleNet consists of 27 layers which explain the higher score it achieves. For experiment 1, with 12 camera models, AlexNet achieves 94.5%, and 91.8% for for $Residual1$, $Residual2$ respectively. GoogleNet achieves 98.99%, and 95.9% for $Residual1$, $Residual2$ respectively. We achieved with 12 camera models, 98% and 95.1% for $Residual1$, $Residual2$ respectively.

The trend is similar for the experiments with 14 camera models. AlexNet achieves 90.5% (respectively 89.45%) for $Residual1$ (respectively $Residual2$). We achieve 97.09% (respectively 93.23%) for $Residual1$ (respectively $Residual2$) and GoogleNet achieves 98.01% (respectively 96.41%) for $Residual1$ (respectively $Residual2$).

We see that our proposition improves AlexNet with 7% for the 14 camera models and the efficiency is only 1% above the bigger network of GoogleNet. As a complexity measure, the time expended for training 12 camera models using our proposed CNN model is about 5 hours and a half, while the time expended for training the same set using GoogleNet is about 16 hours. The time expended by our model for testing 12 camera is about 10 minutes against 30 minutes for GoogleNet. We conclude that our CNN model has good performance for a really smaller complexity compared to GoogleNet.

We should also add that compared to the state of the art approaches based on classical feature extraction and machine learning, the obtained results are similar with a proposition such as [14]. The two methods are implemented in different conditions since the classical machine learning approach [14] uses the full resolution of the data set while the proposed CNN method uses images of size 256×256. GoogleNet gives similar global accuracy (98.99%) with the same set of 14 models. This is thus a good point for CNNs approaches.

By achieving the perfect design of CNNs and well tuning the network we can achieve more than the classical methods listed in the state of the art.

TABLE V
IDENTIFICATION ACCURACIES FOR ALL THE EXPERIMENTS COMPARED TO ALEXNET AND GOOGLENET.

	Exp 1 (1-12) models		Exp 2 (1-14) models		Exp 3 (1-33) models
Method	residual 1	residual 2	residual 1	residual 2	residual 1
AlexNet	94.50%	91.8%	90.50%	89.45%	83.5%
GoogleNet	98.99%	95.9%	98.01%	96.41%	94.5%
Proposed Net	98.00%	95.1%	97.09%	93.23%	91.9%

V. CONCLUSION

In this paper, we evaluate the efficiency of using CNNs for source camera model identification based on deep learning and convolutional neural networks. The contribution represents a big challenge since it is quite different from existing conventional techniques for camera identification. We tried a small net by tuning the AlexNet model. This small network is nevertheless slightly less efficient (1% to 3%) than the biggest GoogleNet model. The varying results with the two different preprocessing filters show the important role that the preprocessing plays in the overall classification accuracy.

Scalability has also been evaluated and the increase of the number of models decreases the accuracy not too drastically. Increasing the number of layers seems to be promising and future work should explore bigger networks such as ResNet of Microsoft [24] (which consists of more than 150 layers).

ACKNOWLEDGMENT

This work was partially funded and supported by the Ministry of Higher Education and Scientific Research in Iraq, Northern Technical University.

REFERENCES

[1] M. Kirchner and T. Gloe, "Forensic camera model identification," in *T. Ho, S. Li, (eds.) Handbook of Digital Forensics of Multimedia Data and Devices. Wiley-IEEE Press*, 2015.

[2] 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, June 2006.

[3] K. Choi, E. Lam, and K. Wong, "Source camera identification using footprints from lens aberration," in *Proc. SPIE, Digital Photography II*, vol. 6069, no. 1, pp. 60 690J, 2006.

[4] A. E. Dirik, H. T. Sencar, and N. Memon, "Source camera identification based on sensor dust characteristics," in *IEEE Workshop on Signal Processing Applications for Public Security and Forensics, SAFE '07, Washington, USA*, pp. 1–6, 2007.

[5] S. Bayram, H. Sencar, and N. Memon, "Improvements on source camera model identification based on cfa interpolation," in *Advances in Digital Forensics II, IFIP International Conference on Digital Forensics, Orlando Florida*, pp. 289–299, 2006.

[6] M. Kharrazi, H. Sencar, and N. Memon, "Blind source camera identification," in *IEEE International Conference on Image Processing ICIP 2004.*, vol. 1, pp. 709–712, 2004.

[7] O. Celiktutan, B. Sankur, and I. Avcibas, "Blind identification of source cell-phone model." *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 553–566, 2008.

[8] T. Filler, J. Fridrich, and M. Goljan, "Using sensor pattern noise for camera model identification," in *Proc. of the 15th IEEE International Conference on Image Processing (ICIP), San Diego, California, October 12-15*, pp. 1296–1299, 2008.

[9] T. Gloe, "Feature-based forensic camera model identification," *Y.Q. Shi, S. Katzenbeisser, (eds.) Transactions on Data Hiding and Multimedia Security VIII. LNCS*, vol. 7228, pp. 42–62, Springer, Heidelberg, 2012.

TABLE III

IDENTIFICATION ACCURACY (IN PERCENTAGE POINTS %) OF THE PROPOSED METHOD FOR $Residual1$, THE TOTAL ACCURACY IS 98%. $-$ MEANS ZERO OR LESS THAN 0.1.

Camera Model		1	2	3	4	5	6	7	8	9	10	11	12
Agfa DC-733s	1	96.35	1.87	-	-	-	-	-	-	-	0.61	0.92	-
Agfa DC-830i	2	2.54	94.5	0.2	0.2	-	-	0.25	-	0.21	1.69	-	-
Agfa Sensor 530s	3	-	-	99.57	-	-	-	0.23	-	-	-	-	-
Canon Ixus 55	4	-	-	-	98.54	-	-	-	-	-	0.89	-	-
Fujifilm FinePix J50	5	-	-	-	-	98.17	-	-	-	-	-	-	0.97
Kodak M1063	6	-	-	-	-	-	99.89	-	-	-	-	-	-
Nikon D200	7	-	-	0.55	-	-	0.21	97.83	0.32	-	-	-	0.61
Olympus M1050	8	-	-	-	-	-	-	0.7	96.38	0.98	-	-	0.9
Panasonic DMC-FZ50	9	-	-	-	-	-	-	-	0.78	98.46		-	0.5
Praktica DCZ 5.9	10	3.91	2.82	-	-	-	-	0.82	-	-	90.44	-	1.83
Samsung L74wide	11	1.1	-	-	-	-	-	-	-	-	0.34	98.13	-
Samsung NV15	12	-	-	-	-	0.93	-	1.21	-	0.62	-	-	96.73

TABLE IV

IDENTIFICATION ACCURACY (IN PERCENTAGE POINTS %) OF THE PROPOSED METHOD FOR $Residual2$, THE TOTAL ACCURACY IS 95.1%. $-$ MEANS ZERO OR LESS THAN 0.1.

Camera Model		1	2	3	4	5	6	7	8	9	10	11	12
Agfa DC-733s	1	85.34	7.64	1.63	-	-	-	-	-	-	2.05	2.52	-
Agfa DC-830i	2	7.54	85.0	1.1	-	-	-	-	-	-	3.23	3.04	-
Agfa Sensor 530s	3	-	0.11	98.09	-	-	0.41	0.11	-	-	-	-	-
Canon Ixus 55	4	-	-	-	98.53	-	0.32	-	-	-	0.75	-	-
Fujifilm FinePix J50	5	-	-	0.11	-	99.25	0.13	-	-	-	-	-	-
Kodak M1063	6	-	-	0.15	-	-	99.2	0.19	-	-	-	-	-
Nikon D200	7	-	-	0.56	-	-	0.74	97.5	-	-	0.12	-	0.46
Olympus M1050	8	-	0.42	-	-	-	-	0.15	97.15	-	0.1	-	0.22
Panasonic DMC-FZ50	9	0.18	-	-	-	-	-	-	-	99.46		-	-
Praktica DCZ 5.9	10	6.36	3.96	2.1	0.13	-	0.93	0.74	-	-	81.54	-	3.29
Samsung L74wide	11	2.26	1.1	0.97	-	-	0.1	-	-	-	-	89.81	5.12
Samsung NV15	12	0.49	0.18	0.14	-	-	0.23	0.14	-	-	0.41	-	97.92

[10] G. Xu and Y. Q. Shi, "Camera model identification using local binary patterns," in *Proc. IEEE Int Conference on Multimedia and Expo (ICME), Melborne, Australia*, pp. 392–397, 2012.

[11] A. AbdulWahab, A. Ho, and S. Li, "Inter camera model image source identification with conditional probability features," in *Proc. of the 3rd Image Electronics and Visual Computing Workshop*, 2012.

[12] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Evaluation of residual-based local features for camera model identification," in *New Trends in Image Analysis and Processing - ICIAP Workshop : BioFor, Genoa, Italy, September 7-8*, pp. 11–18, 2015.

[13] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, June 2012.

[14] A. Tuama, F. Comby, and M. Chaumont, "Source camera identification using features from contaminated sensor noise," *IWDW 2015, The 14th International Workshop on Digital-forensics and Watermarking, Proceedings as Lecture Notes in Computer Science (LNCS) by Springer, Tokyo, Japan, 7-10 october, 11 pages, 2015.*

[15] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, ser. IH&MMSec'16. Vigo, Galicia, Spain: ACM, 2016.

[16] J. Chen, X. Kang, Y. Liu, and Z. Wang, "Median filtering forensics based on convolutional neural networks," *Signal Processing Letters, IEEE*, vol. 22, no. 11, pp. 1849–1853, Nov 2015.

[17] M. Najafabadi, F. Villanustre, T. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Springer*, vol. 2, 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*. Curran Associates Inc., pp. 1097–1105, 2012.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA, June 7-12, pp. 1-9, 2015.*

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Analysis and Machine Intellegence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[21] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Proc. SPIE*, vol. 9409, pp. 94 090J–94 090J–10, 2015.

[22] J. Fridrich, "Digital image forensic using sensor noise," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26–37, 2009.

[23] T. Gloe and R. Böhme, "The 'Dresden Image Database' for benchmarking digital image forensics," in *Proceedings of the 25th Symposium On Applied Computing (ACM SAC 2010)*, vol. 2, pp. 1585–1591, 2010.

[24] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition," *Technical Report*, 2015.