TOWARDS OPEN SET CAMERA MODEL IDENTIFICATION USING A DEEP LEARNING FRAMEWORK

Belhassen Bayar and Matthew C. Stamm

Department of Electrical and Computer Engineering,
Drexel University, Philadelphia, PA 19104

ABSTRACT

Existing forensic camera model identification algorithms can be trained to accurately distinguish between a set of known camera models. In reality, however, an investigator may be confronted with an image that was not captured by one of these known models. If this happens, existing algorithms will associate this image with one of the known camera models. This is known as the open set problem. In this paper, we propose two different approaches to address the open set problem for camera model identification. To do this, we use a CNN to learn a set of deep forensic features. Our first approach replaces the CNN's classifier with a confidence score mapping which it thresholds to detect unknown models. Our second approach uses a set of 'known unknown' models to train a new classifier to identify unknown camera models. Experiments show that we can detect unknown camera models with a 97.74% accuracy.

Index Terms— Open set problem, camera model identification, deep convolutional features.

1. INTRODUCTION

Camera model identification is an important problem in multimedia forensics. Information about the make and model of an image's source camera can be used in many important settings, such as evidence in legal proceedings and criminal investigations. While metadata can contain this information, it is frequently missing and can be easily altered. Therefore, forensic researchers have developed methods to 'blindly' determine an image's source camera model by identifying its fingerprints left in a captured image [1].

These fingerprints are unique from one camera model to another and are specifically linked to signal processing artifacts induced by different components of a camera's internal processing pipeline. Early approaches used heuristically designed statistical metrics as features to measure and determine camera's traces [2]. Other techniques use specific physical component traces such as the camera's imaging sensor [3, 4]. Many existing methods rely on the algorithmic components such as the unique implementation of JPEG compression [5] and traces left by demosaicing [6, 7, 8, 9]. These existing techniques are often designed by constructing local parametric models of an image's data [6, 7] or utilize hand-designed features [10, 11].

Recent work in multimedia forensics suggests that learning camera's features can be accomplished by using convolutional neural networks (CNNs) [12, 13, 14]. The advantage of CNNs is that they are capable of learning classification features directly from data, hence, they can adaptively learn the cumulative traces induced by a camera's components. However, CNNs in their existing form tend to learn features related to an image's content. In response to this issue, two alternatives have emerged to suppress an image's content and capture camera's forensic features, i.e., high-pass filter (HPF) [13] and the adaptive constrained convolutional layer [12].

While existing forensic methods have shown great promise [12, 14, 11, 10], these methods cannot perform camera model identification to images taken by '*unknown*' camera models not used to train the forensic detector. This is known as the open set problem. Open set problems have been studied in different applications such as computer vision [15], fingerprint spoof detection [16], source camera attribution [17] (i.e., specific device, not camera model), etc.

If a traditional forensic method is used to identify an unknown image's source camera model, this will lead to mistakenly associate the subject image with one of the known camera models used for the training. This is problematic since in real world scenario a forensic investigator has at their disposal a limited number of camera models to train a forensic detector. Therefore, one may ask: can we learn forensic detection features to perform camera model identification in an open set scenario? Can we devise a reliable forensic protocol that can determine how likely a testing sample belongs to one of the known classes?

In this paper, we propose a new deep learning method to address the open set camera model identification problem. To accomplish this, we devised two forensic protocols that account for two different open set scenarios. Both protocols, are associated with a constrained CNNs [12, 18] to extract deep forensic features from known and unknown camera models. In the first protocol, the learned deep forensic features will go through a function that produces a confidence score of how likely a subject image was taken by each of the known camera models used for training. Next, a thresholding protocol is used on the maximum confidence score over all the known classes, to identify unknown camera models. In our second protocol we used a set of '*known unknown*' camera models to build a classifier to discriminate between deep forensic features of images taken by known and unknown cameras collected from an external database. Through a set of experiments, we demonstrate that our two proposed protocols can achieve strong identification rates in scenarios where the set of unknown models is larger than the set of known models.

2. PROBLEM FORMULATION

The goal of this paper is to develop and evaluate methods to determine if an image was taken by a camera within a set of known camera models $\mathcal{T}$ or if it was taken by an unknown model.

To frame this problem, let us first consider the standard camera model identification scenario where a forensic investigator wants to

This material is based upon work supported by the National Science Foundation under Grant No. 1553610. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

design some system $g(\cdot)$ that identifies the model of the camera that captured some input image or image patch $\boldsymbol{x}$. In general, this system g can be thought of as the composition of two functions, $f(\cdot)$ and $d(\cdot)$, i.e.

$$g(\boldsymbol{x}) = d \circ f(\boldsymbol{x}) = d(f(\boldsymbol{x})), \tag{1}$$

where $f(\cdot)$ is a feature extractor and $d(\cdot)$ is a classifier that discriminates between a set of candidate camera models $\mathcal{T}$ and $\circ$ denotes function composition.

The classifier d and potentially the feature extractor f must be learned from a set of labeled data $\mathcal{D}$ which corresponds to a set of images taken by each of the camera models in $\mathcal{T}$. Significant prior research has shown that an accurate camera model identification algorithm $g(\cdot)$ can be built to identify the source camera model of images taken by cameras in $\mathcal{T}$. In practice, however, the set of camera models $\mathcal{T}$ that the forensic investigator has access to is a subset of the set of all camera models $\mathcal{M}$, i.e. $\mathcal{T} \subset \mathcal{M}$. The number of camera models in $\mathcal{T}$ may be significantly smaller than the number of models in $\mathcal{M}$.

This gives rise to an important problem: in reality an investigator can't guarantee that all images whose source they wish to identify came from some known model $m \in \mathcal{T}$. It is very possible that the true camera model $m \in \mathcal{T}^c$, i.e. the image under investigation comes from some "unknown" camera model in $\mathcal{M}$. The camera model identification algorithm, specifically the classifier d, must choose only between "known" camera models in $\mathcal{T}$, even if the true source camera model is in $\mathcal{T}^c$. This problem is known as the "open set problem".

In order for forensic investigators to trust the camera identification results provided by g, they need some means of first determining if the true source camera model is in the set of known camera models used for the training, i.e., $m \in \mathcal{T}$.

3. PROPOSED METHOD

This paper proposes two strategies for addressing the open set problem for camera model identification.

While the strategies we propose can be easily generalized to other camera model identification approaches, in this paper we assume g is a CNN-based camera model identification algorithm. Specifically, we define g as the constrained CNN that we previously proposed in [12] but using hyperbolic tangent (TanH) activation functions instead of ReLUs. We choose g to be this CNN because it is very important for the feature extractor f to extract highly discriminant camera model identification features, since these features must be relied upon to differentiate between camera models in $\mathcal{T}$ and $\mathcal{T}^c$. Recent research suggests that CNNs are able to learn these highly discriminant features [12, 13, 14].

Using an approach that is well known in computer vision [19], we define the feature extractor $f(\cdot)$ portion of our CNN as all of the CNN's layers which precede the classification layer, as we have previously done in [20, 21, 22]. After the CNN is trained to distinguish between the N camera models in $\mathcal{T}$, feature values for an image or image patch $\boldsymbol{x}$ can be obtained by passing $\boldsymbol{x}$ through the CNN. The resulting feature values $f(\boldsymbol{x})$, which we refer to as *deep forensic features*, correspond to the neuron activations of the second-to-last fully connected layer (denoted in [20] as "FC_2") of the CNN.

3.1. Approach 1

Our first approach for performing open set camera model identification is called "confidence score thresholding." In this approach, we first build and train a camera model identification system (i.e. CNN) g using training data $\mathcal{D}$ from each of the N camera models in the set of known models $\mathcal{T}$. We retain the feature extractor f learned while training g, but set aside the classifier d. We replace d with a new classifier d' composed of a mapping $c(\cdot)$ designed to produce a confidence score associated with each camera model $m_k \in \mathcal{T}$ and a protocol for choosing a class on the basis of these scores. We note that while a CNN with a softmax layer can produce confidence scores, our experimental results shown in Sec. 4 will demonstrate that this is a suboptimal approach.

The confidence score mapping takes the deep forensic feature vector $f(\boldsymbol{x})$ learned by g as an input and produces a scalar output. In practice, this can be accomplished by using $\mathcal{D}$ to train a new classifier to produce confidence scores, or by using a metric to measure the distance from the mean value of the deep forensic features associated with each camera model. Camera model identification is then performed by taking the $\arg\max$ of all of these confidence scores. Let $c_k\big(f(\boldsymbol{x})\big)$ be the score associated with camera model $m_k \in \mathcal{T}$. The new camera model identification system g' formed is

$$g'(\boldsymbol{x}) = \arg\max_{m_k \in \mathcal{T}} c_k\big(f(\boldsymbol{x})\big) = \hat{m}, \tag{2}$$

where $\hat{m} \in \mathcal{T}$ is the camera model that this system identifies as the source of $\boldsymbol{x}$ and

$$s = \max_{m_k \in \mathcal{T}} c_k\big(f(\boldsymbol{x})\big), \tag{3}$$

is the confidence score associated with this decision.

Now the investigator must decide whether to accept the camera model $\hat{m}$ identified by g' (with the implicit assumption that the true source model $m \in \mathcal{T}$) or to reject $\hat{m}$ with the belief that $m \notin \mathcal{T}$ (i.e., the image $\boldsymbol{x}$ was captured by an unknown camera model). To make this decision, we pose this problem as the following hypothesis testing problem

$$\begin{aligned} H_0 &: \text{ The true source camera model is known, i.e., } m \in \mathcal{T}, \\ H_1 &: \text{ The true source camera model is unknown, i.e., } m \notin \mathcal{T}. \end{aligned} \tag{4}$$

We differentiate between these hypotheses using the following decision rule h_1

$$h_1(s) = \begin{cases} H_0, & s \geq \eta \\ H_1, & s < \eta. \end{cases} \tag{5}$$

where scalar η is a decision threshold. If $h_1(s)$ returns H_0, then the camera model $\hat{m}$ identified by d' is accepted. If $h_1(s)$ returns H_1, then the camera model $\hat{m}$ identified by d' is rejected and the investigator concludes that the true source camera model is unknown (i.e., $m \notin \mathcal{T}$).

Confidence Score Choices: In this work, we examine four different choices for $c(\cdot)$: (1) one fully connected layer neural network followed by a softmax (i.e. the baseline choice), (2) an extremely randomized trees (ET) classifier trained to produce confidence scores [23], (3) a multi-class P-SVM [24] with a radial basis function (RBF) kernel, and (4) the cosine similarity measure distance from the nearest class mean. These are briefly described below.

Confidence scores produced by the softmax approach $c_j^{(1)}$ are defined as follows. Let $v(f(\boldsymbol{x}))$ be the vector of activation values of the additional fully connected layer and $v(f(\boldsymbol{x}))_j$ be the activation value of the j^{th} neuron. The softmax score c_j assigned to the j^{th} class (i.e. camera model) is defined as

$$c_j^{(1)}\big(f(\boldsymbol{x})\big) = c_j\big(f(\boldsymbol{x})\big) = \frac{e^{v(f(\boldsymbol{x}))_j}}{\sum_{i=1}^{N} e^{v(f(\boldsymbol{x}))_i}} \tag{6}$$

Confidence scores $c_j^{(2)}$ produced by the extremely randomized tree approach are defined as the mean of the classification probabilities of all the trees that vote for the j^{th} camera model. The classification probability of each tree is the fraction of samples of the training data $\mathcal{D}$ correspoding to the j^{th} camera model that are correctly classified by each leaf in the tree [23].

Confidence scores produced by the P-SVM approach $c_j^{(3)}$ correspond to the multi-class extension [24] of an SVM with Platt Scaling [25]. In this approach, the training data $\mathcal{D}$ is used to fit a logistic sigmoid to the decision boundaries of the SVM in order to produce scores.

We produce confidence scores using the cosine similarity metric $c_j^{(4)}$ by first using the labeled training data $\mathcal{D}$ to compute the mean value of the deep feature vector $\boldsymbol{\mu}_j$ corresponding to each camera model $m_j \in \mathcal{T}$. The confidence score assigned to the j^{th} camera model corresponds to the cosine similarity distance from $\boldsymbol{\mu}_j$ such that

$$c_j^{(4)}\big(f(\boldsymbol{x})\big) = \cos\text{sim}\left(\boldsymbol{\mu}_j, f(\boldsymbol{x})\right) = \frac{\boldsymbol{\mu}_j \cdot f(\boldsymbol{x})}{\|\boldsymbol{\mu}_j\|_2 \|f(\boldsymbol{x})\|_2} \quad (7)$$

In this case, it should be noted that the confidence score s is defined in the interval $[-1, 1]$ as opposed to approaches (1)-(3) which produce a confidence score in the interval $[0, 1]$.

3.2. Approach 2

Our second approach builds a separate 'known vs. unknown' classifier using the deep features extracted by f. This approach is appropriate for a scenario where a forensic analyst has access to a larger set of known camera models $\mathcal{T}$ but only needs to identify a subset of these models. In such scenario, one can use some of these models to learn decision boundaries that separate the feature distributions of known and unknown cameras. To accomplish this, we use the definition of '*known unknown*' cameras to partition $\mathcal{T}$ into two disjoint sets of known camera models, i.e., 'knowns' and 'known unknowns'.

We partition the camera model set $\mathcal{T}$ into two disjoint subsets $\mathcal{T}_\mathcal{A}$ and $\mathcal{T}_\mathcal{B}$, where $\mathcal{T}_\mathcal{A}$ is the set of 'knowns' and $\mathcal{T}_\mathcal{B}$ corresponds to the set of 'known unknowns'. This scenario could be seen as a 'closed open set problem'. Next, we gather a labeled set of training data $\mathcal{D}$ using all cameras in $\mathcal{T}$. We define $\mathcal{D}_\mathcal{A}$ as the set of data collected by cameras in $\mathcal{T}_\mathcal{A}$, and $\mathcal{D}_\mathcal{B}$ as the set of data collected by cameras in $\mathcal{T}_\mathcal{B}$. We further partition $\mathcal{D}_\mathcal{A}$ into two disjoint subsets $\mathcal{D}_\mathcal{A}^{(1)}$ and $\mathcal{D}_\mathcal{A}^{(2)}$ which both contain images from all models in $\mathcal{T}_\mathcal{A}$.

Our approach proceeds by first training the CNN g defined in Sec. 2 using data from $\mathcal{D}_\mathcal{A}^{(1)}$ to distinguish between models in $\mathcal{T}_\mathcal{A}$. we retain the feature extractor f and we build a new classifier $h_2(\cdot)$ designed to distinguish between known models (i.e., $m \in \mathcal{T}_\mathcal{A}$) and unknown models (i.e., $m \notin \mathcal{T}_\mathcal{A}$). We use $\mathcal{T}_B$ which is the set of 'known unknowns' to represent models in ${\mathcal{T}_\mathcal{A}}^c$.

We extract features from images $\boldsymbol{x}$ using the pre-trained feature extractor f. These are passed to a new classifier h_2 which is trained to distinguish between known and unknown camera models. When training h_2, dataset $\mathcal{D}_\mathcal{A}^{(2)}$ is used for training data for known camera models and data set $\mathcal{D}_\mathcal{B}$ is used to represent unknown models. By using $\mathcal{D}_\mathcal{A}^{(2)}$ to train this classifier, we can avoid overfitting that may occur if data from $\mathcal{D}_\mathcal{A}^{(1)}$ was used (since $\mathcal{D}_\mathcal{A}^{(1)}$ was used to learn the feature extractor).

It is important to note that h_2 only chooses between two classes: known models and unknown models. If $h_2(\boldsymbol{x})$ yields a decision that $\boldsymbol{x}$ originates from a known model (i.e., $m \in \mathcal{T}_\mathcal{A}$), then g can be used to identify this model. In this work, we consider the following two classifiers for $h_2(\cdot)$: a binary SVM with an RBF Kernel and Platt's probability estimator [25] and a binary ET classifier.

As our experimental results presented in Section 4 will show, this approach has the advantage of being able to more accurately distinguish between known and unknown camera models. It comes, however, with a disadvantage in that g is only able to distinguish between models in $\mathcal{T}_\mathcal{A}$ instead of all models in $\mathcal{T}$. One way to overcome this, is one can define $\mathcal{T}_\mathcal{A} = \mathcal{T}$ then use our approach in Section 3.1 to collect known unknowns data $\mathcal{D}_\mathcal{B}$ (i.e., $m \in \mathcal{T}_\mathcal{B}$) from the 'wild' based on the decisions made by h_1 in Eq. (5). Next, the classifier h_2 can be trained with features extracted by $f(\boldsymbol{x})$ from images taken by $\mathcal{T}_\mathcal{A}$ and $\mathcal{T}_\mathcal{B}$ to distinguish between known and unknown models.

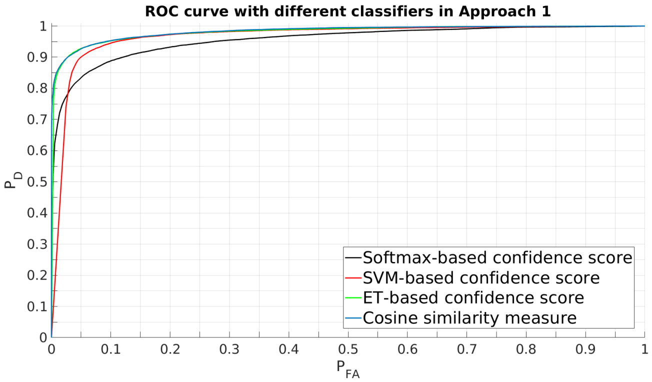

Fig. 1: ROC curves for Approach 1 with different classifiers

4. EXPERIMENTS

We conducted a set of experiments to evaluate the performance of our two proposed approaches at performing camera model identification in an open set scenario. In these experiments, we first used Approach 1 to identify unknown camera models, which accounts for the scenario where a forensic investigator must use the whole set of camera models $\mathcal{T}$ to perform open set camera model identification. Next, we used Approach 2 to detect unknown camera models using a larger set of training camera models that we can partition into two set of 'known' and 'known unknown' camera models. To extract deep forensic features $f(\boldsymbol{x})$, we trained our previously proposed CNN [12] with TanH activation functions in both approaches.

During the training phase of the CNNs, we set the batch size equal to 64 and the parameters of the stochastic gradient descent as follows: $momentum = 0.9$, $decay = 0.0005$, and a learning rate $\epsilon = 10^{-3}$ that decreases every 4 epochs by a factor $\gamma = 0.5$. In our experiments, we trained the CNN for 44 epochs. CNNs were built using Caffe [26] and trained on an Nvidia GTX 1080Ti GPU.

4.1. Approach 1 evaluation

First, we created our set of known camera models $\mathcal{T}$ by collecting images captured by 10 different camera models from the Dresden Image Database [27]. Our first approach proceeds by training our CNN as a classifier g to distinguish between 10 camera models in $\mathcal{T}$. To do this, we created $90,000$ grayscale 256×256 training patches using $2,500$ images captured by 10 camera models in $\mathcal{T}$.

These training grayscale patches were selected from the green layer of the central 36 blocks of each image. Each patch corresponds to a new image associated with one camera model class. We chose a small number of camera models in order to mimic the real world scenario where $|\mathcal{T}^c| > |\mathcal{T}|$.

Next, we trained different classifiers d' using the training deep forensic features $f(\boldsymbol{x})$ extracted from CNN to discriminate between camera models in $\mathcal{T}$. To do this, we used (1) the softmax classification layer of our previously trained CNN, (2) an ET classifier, (3) a RBF multi-class SVM [24], and (4) a cosine similarity measure nearest mean score. We tuned the RBF parameters for SVM

Table 1: Known vs. unknown detection accuracy and closed set testing accuracy with different classifiers in Approach 1.

Method	**Detection accuracy**	**Accuracy (closed set)**
ET	**93.93%**	**99.06%**
Softmax	89.53%	98.50%
SVM	92.72%	99.05%
Cos Sim	93.84%	98.70%

Table 2: Known vs. unknown detection accuracy in Approach 2.

	Classifiers	
Testing datasets	**SVM**	**ET**
$\mathcal{D}_1 : m \in \mathcal{T} = \{\mathcal{T}_{\mathcal{A}}, \mathcal{T}_{\mathcal{B}}\}$	99.20%	99.38%
$\mathcal{D}_2 : m \in \{\mathcal{T}, \mathcal{T}^c\}$	98.27%	98.57%
$\mathcal{D}_3 : m \in \{\mathcal{T}_{\mathcal{A}}, \mathcal{T}^c\}$	97.37%	97.74%

via 5-fold cross validation on $20,000$ randomly selected deep features of training patches using a grid of $C = 2^{-5}, 2^{-3}, \cdots, 2^{11}$ and $\gamma = 2^{-15}, 2^{-13}, \cdots, 2^{3}$. To find the best parameters of the ET classier we used a grid search over the number of trees in the forest for values $100, 200, \cdots, 800$.

To evaluate our proposed approach, we created $46,000$ grayscale testing patches of known and unknown camera models in the same manner we described above. In total, we used 639 images from the Dresden database and 639 images captured by 15 unknown camera models from our lab experimental database. Note that training and testing datasets were created from two separate sets of images.

Next, we computed the known vs. unknown detection accuracy of the Bayes rule in Eq. (5), the probability of false alarm (P_{FA}), and the probability of detection P_D using our testing dataset for different values of the arbitrary threshold. We also computed the testing accuracy in the closed set scenario with only known camera models. In Table 1, we report the known vs. unknown detection accuracy as the *max* over all testing rates obtained by different values of η as well as the testing accuracy with 10 camera models in a closed set scenario.

From Table 1, we can notice that the ET based approach outperforms the other choices of classifiers in the above two mentioned tasks. Noticeably, the ET based approach can achieve 93.93% known vs. unknown detection accuracy, and 99.06% testing accuracy in a closed set scenario. We can also notice that one can significantly improve over the known vs. unknown detection accuracy of a softmax-based CNN by using the deep forensic features to train other discriminative classifiers which learn better decision boundaries. Moreover, these results demonstrate that a constrained CNN can learn forensic features to detect unknown camera models even from images captured by camera models not used for the training. Note that an appropriate choice of the confidence score mapping $c(\cdot)$ can result in significantly better known vs. unknown detection accuracy and also improves the closed set accuracy.

Fig. 1 depicts the ROC curve for the different discriminative classifiers. One can observe that the ET and cosine similarity based threshold techniques can achieve comparable performance and outperform the other choices of classifiers. Moreover, for $P_{FA} < 5\%$ the ET and cosine similarity approaches can achieve at least 9% better P_D than the softmax and SVM outperforms the softmax detection rate with about 7%.

4.2. Approach 2 evaluation

Next, we considered that a forensic investigator has access to a larger set of camera models so that they can make a disjoint partition of $\mathcal{T}$ into known ($m \in \mathcal{T}_{\mathcal{A}}$) and known unknown ($m \in \mathcal{T}_{\mathcal{B}}$) camera models. To do this, we used our previously defined set of 10 camera models from Dresden as $\mathcal{T}_{\mathcal{A}}$. Then we used 15 different camera models from our lab database to define $\mathcal{T}_{\mathcal{B}}$. Subsequently, we created a dataset $\mathcal{D}_{\mathcal{A}}^{(1)}$ to train a CNN g. Next, we created $\mathcal{D}_{\mathcal{A}}^{(2)}$ and $\mathcal{D}_{\mathcal{B}}$ to train h_2 in order to distinguish between known and unknown models as described in Section 3.2.

To accomplish this, we created $\mathcal{D}_{\mathcal{A}}^{(1)}$ and $\mathcal{D}_{\mathcal{A}}^{(2)}$ by evenly dividing the Dresden training dataset that we created for Approach 1 into $45,000$ patches for each. Next, we created $\mathcal{D}_{\mathcal{B}}$ that consisted of $45,000$ patches by using $1,250$ images captured by 15 known unknown camera models from our lab experimental dataset. In this work, we examined two different choices of binary classifiers for h_2, i.e., ET [28] and SVM [25]. We tuned the parameters of the binary classifiers similarly to Approach 1's experiments. Finally, we created three different testing datasets (i.e., $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$) that we describe below.

We created three testing datasets where each consisted of $46,000$ grayscale 256×256 testing patches in the same manner described above. First, $\mathcal{D}_1$ consisted of 'known' and 'known unknown' patches where $23,000$ patches were created from 639 images captured by 10 known camera models from Dresden and $23,000$ patches were created from 639 images captured by 15 known unknown camera models from our lab database. Next, we created $23,000$ grayscale testing patches from 639 images captured by another new 15 unknown camera models from our lab database where these camera models never been used to train any classifier, i.e., $m \in \mathcal{T}^c$. Our $\mathcal{D}_2$ dataset consisted of 'known', 'known unknown' and 'unknown' patches where we used the same $23,000$ patches in $\mathcal{D}_1$ captured by 10 known camera models, $11,500$ patches from 15 known unknown camera models, and $11,500$ patches from 15 completely unknown camera models never used for the training. $\mathcal{D}_3$ consisted of 'known' and 'unknown' patches where we used the same $23,000$ patches captured by 10 known camera models from Dresden and the $23,000$ patches captured by 15 unknown camera models never used for the training. Note that none of the testing patches was created from images used to create training patches.

In Table 2, we report the unknown camera models detection rate using our proposed binary classifiers. One can notice that our ET-based approach associated with the deep forensic features outperforms the choice of SVM classifier. Noticeably, it can achieve 99.38% accuracy with $\mathcal{D}_1$, 98.57% accuracy with $\mathcal{D}_2$, and 97.74% accuracy with $\mathcal{D}_3$. Furthermore, our experimental result on $\mathcal{D}_3$ demonstrates that one can significantly improve over our "confidence score thresholding" approach using the set of known unknown camera models $\mathcal{T}_{\mathcal{B}}$.

5. CONCLUSION

In this paper, we have proposed two approaches to perform camera model identification in open set scenarios. To accomplish this, we first used a constrained CNN to extract camera model identification features. Our first approach proceeds by mapping the learned deep features onto confidence scores associated with each known camera model used to train the CNN. A thresholding protocol over the maximum confidence score was used to identify unknown cameras. Finally, we used the definition of known unknowns to represent the set of real unknown camera model. Then, we used two different binary classifiers to discriminate between deep features of the known and unknown camera models. Through a set of experiments, we demonstrated the effectiveness of our approach using an external camera models database.

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

[3] T. Filler, J. Fridrich, and M. Goljan, “Using sensor pattern noise for camera model identification,” in *the 2008 IEEE International Conference on Image Processing*. IEEE, 2008, pp. 1296–1299.

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

[5] E. Kee, M. K. Johnson, and H. Farid, “Digital image authentication from jpeg headers,” *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, Sep. 2011.

[6] A. Swaminathan, M. Wu, and K. J. 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.

[7] 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.

[8] S. Milani, P. Bestagini, M. Tagliasacchi, and S. Tubaro, “Demosaicing strategy identification via eigenalgorithms,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2659–2663.

[9] X. Zhao and M. C. Stamm, “Detecting anti-forensic attacks on demosaicing-based camera model identification,,” in *IEEE International Conference on Image Processing*, Beijing, China, Sep. 2017.

[10] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, “A study of co-occurrence based local features for camera model identification,” *Multimedia Tools and Applications*, pp. 1–17, 2016.

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

[12] B. Bayar and M. C. Stamm, “Design principles of convolutional neural networks for multimedia forensics,” in *International Symposium on Electronic Imaging: Media Watermarking, Security, and Forensics*. IS&T, 2017.

[13] A. Tuama, F. Comby, and M. Chaumont, “Camera model identification with the use of deep convolutional neural networks,” in *IEEE International Workshop on Information Forensics and Security*, 2016, pp. 6–pages.

[14] 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.

[15] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult, “The extreme value machine,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[16] A. Rattani, W. J. Scheirer, and A. Ross, “Open set fingerprint spoof detection across novel fabrication materials,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2447–2460, 2015.

[17] F. d. O. Costa, M. Eckmann, W. J. Scheirer, and A. Rocha, “Open set source camera attribution,” in *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on*. IEEE, 2012, pp. 71–78.

[18] 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*. ACM, 2016, pp. 5–10.

[19] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *ICML*, 2014, pp. 647–655.

[20] B. Bayar and M. C. Stamm, “Augmented convolutional feature maps for robust cnn-based camera model identification,” in *IEEE International Conference Image Processing*, Beijing, China, Sep. 2017.

[21] B. Bayar and M. C. Stamm, “Towards order of processing operations detection in jpeg-compressed images with convolutional neural networks,” in *International Symposium on Electronic Imaging: Media Watermarking, Security, and Forensics*. IS&T, 2018.

[22] B. Bayar and M. C. Stamm, “On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection,” in *The 2017 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017.

[23] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[24] T.-F. Wu, C.-J. Lin, and R. C. Weng, “Probability estimates for multi-class classification by pairwise coupling,” *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 975–1005, 2004.

[25] J. Platt *et al.*, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *arXiv preprint arXiv:1408.5093*, 2014.

[27] 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.

[28] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.