DETECTING ANTI-FORENSIC ATTACKS ON DEMOSAICING-BASED CAMERA MODEL IDENTIFICATION

Chen Chen, Xinwei Zhao* and Matthew C. Stamm*

Dept. of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104, USA

ABSTRACT

Many forensic algorithms have been developed to determine the model of an image's source camera by examining traces left by the camera's demosaicing algorithm. An anti-forensic attacker, however, can falsify these traces by maliciously using existing forensic techniques to estimate one camera's demosaicing filter, then use these estimates to re-demosaic an image captured by a different camera. Currently, there is no known defense against this attack, which is capable of fooling existing camera model identification algorithms. In this paper, we propose a new method to detect if an image's source camera model has been anti-forensically falsified. Our algorithm operates by characterizing the different content-independent local pixel relationships that are introduced by both authentic demosaicing algorithms and anti-forensic attacks. Experimental results show that our algorithm can detect an anti-forensic attack with over 99% accuracy, is robust to JPEG compression, and can even identify the true source camera model in certain circumstances.

Index Terms— Source Camera Model Falsification, Anti-Forensics, Multimedia Forensics, Demosaicing Traces, Geometric Co-occurrence Patterns

1. INTRODUCTION

Since digital images play critical roles in settings such as news reporting, criminal investigations, and juridical proceedings, it is very important to verify an image's source. While an image's metadata may contain information about its source device, metadata is frequently missing and can easily be modified. As a result, source camera model identification has become an important topic in the field of information forensics [1]. A variety of forensic algorithms have been proposed to idenitify the make and model of an image's source camera using traces such as JPEG header information [2] and sensor noise [3]. Utilizing traces left by an camera's demosaicing algorithm, however, remains one of the most popular and successful approaches [4, 5, 6, 7, 8, 9, 10, 11, 12].

In some scenarios, an attacker may attempt to falsify the source camera model of an image. In this paper, we focus on the anti-forensic falsification of demosaicing traces to acheive this goal. It is well known that an attacker can maliciously use existing forensic methods to obtain estimates of a camera's demosaicing filter, then use these estimates to re-demosaic an image captured by another camera [13, 14]. This attack remains the state-of-the-art, and can effectively falsify a camera's demosaicing traces and fool existing camera model identification algorithms.

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.

Email: {cc3359, xz355}@drexel.edu, mstamm@coe.drexel.edu

* These authors contributed equally to this manuscript.

Currently, there is no known method of detecting this anti-forensic attack. Previous work, however, has shown that other anti-forensic attacks can leave behind their own forensically detectable traces [15, 16, 17, 18, 19]. As a result, it is critical for forensic investigators to identify indicators that demosaicing traces have been falsified and search for any remaining traces left by the camera. In particular, these indicators are likely detectable when examining cross-color channel pixel dependencies and nonlinear in-channel pixel relationships that anti-forensic attacks may not be able to falsify.

In this paper, we propose a new method to detect if an image's source camera model has been falsified by anti-forensically attacking its demosaicing traces. Our algorithm operates by characterizing the different content-independent local pixel relationships that are introduced by both authentic demosaicing algorithms and anti-forensic attacks. To do this, we first gather a set of both linear and nonlinear demosaicing residuals by re-demosaicing the image, then subtracting each re-demosaiced version from the original image. We then form co-occurrence matrices from these residuals using a set of newly designed co-occurrence patterns. These co-occurrence patterns are designed to be both CFA-aware and to capture both inter-channel and intra-channel relationships that anti-forensic attacks will have a difficult time falsifying.

In addition, we propose two new protocols for designing and training an ensemble classifier to detect attacks using these features based on different amounts of knowledge available to the forensic investigator. If the investigator has access to all camera models available to an attacker, we show that our algorithm can both detect an anti-forensic attack and determine the true source camera model with 99.90% accuracy. If, on the other hand, an attacker captures an image using a camera model unavailable to the investigator then falsifies its source, we show that our algorithm can still successfully detect the attack with 99.29% accuracy. Additionally, we show that we are still able to detect an attack even if the image is re-compressed.

2. PROBLEM FORMULATION

Forensic Camera Model Identification: Forensic camera model identification algorithms operate by searching for traces left in an image by different components of a camera's image processing pipeline, i.e. the sequence of physical and algorithmic components that a digital camera uses to form an image [1]. When a digital camera captures a real scene, light L reflected from an object first passes through the lens, then through a color filter array (CFA), before reaching the sensor. The CFA consists of a repeating fixed pattern of three color components (red, green, and blue). Since most sensors are only capable of recording one color band of the light at each pixel location, the CFA only allows one color component to pass through at each pixel location. Consequently, the sensor records the light intensity according to the CFA pattern and yields an image S

composed of three partially sampled color channels. Next, the missing color values in the partially sampled image are filled in through a process known as demosaicing. After demosaicing, the image may undergo post-processing operations such as white balancing or JPEG compression before the final image I is output by the camera.

Since demosaicing has a significant effect on an image's visual quality, most camera manufacturers develop or license proprietary demosaicing algorithms for use in their digital cameras. As a result, demosaicing traces are frequently used to identify a the make and model of an image's source camera [1].

One common approach involves building a linear parametric model of a camera's demosaicing algorithm, estimating the demosaicing filter, then using these estimates as features for camera model identification [4, 5, 7]. The demosaicing filter estimation portion of these algorithms can be viewed as a mapping $\phi(\cdot)$ from the set of images $\mathcal{I}$ to the set of all possible demosaicing filters Θ, i.e. $\phi : \mathcal{I} \rightarrow \Theta$. The classifier that uses these estimates as features to identify a camera model can be viewed as a mapping $f : \Theta \rightarrow \Gamma$ where Γ is the set of all possible camera models. As a result, the entire forensic algorithm m can be viewed as the composition of these two functions such that $m(I) = (f \circ \phi)(I)$. While several successful nonparametric approaches have recently been proposed [6, 9, 10, 11], the parametric approaches described above can be maliciously used by an attacker to launch a general anti-forensic attack [13].

Anti-Forensic Camera Model Falsification: In some scenarios, an attacker may wish to falsify the source of an image. In this paper, we assume an attacker *Alice* wants to fool an investigator *Bob* using an anti-forensic attack $\alpha(\cdot)$. Her attack is designed to make an image I taken by camera model $\gamma^{(k)}$ appear as it it was taken by camera model $\gamma^{(\ell)}$. To accomplish this attack, Alice will attempt to falsify the demosaicing traces that Bob will use to identify the model of the camera that captured the image.

We define the true camera model as the model of the camera that actually took the image, and the target camera model as the one that Alice wants I to appears as if it was taken by. Additionally, we adopt the notation that $I^{(k)}$ denotes an unaltered image whose true camera model is $\gamma^{(k)}$, and $\tilde{I}^{(k,\ell)}$ denotes an anti-forensically modified image whose true camera model is $\gamma^{(k)}$ and whose target camera model is $\gamma^{(\ell)}$. To mimic real scenarios, we assume that Bob may only have access to a subset Ω of the set of all possible source camera models, i.e. $\Omega \subseteq \Gamma$. By contrast, Alice may wish to falsify the origin of images from a camera model that Bob does not have access to (i.e Alice's camera model may be in Ω^c). Since Alice's goal is to make Bob believe the falsified traces she inserts, we assume that Ω forms the set of target camera models.

We assume that camera manufactures keep their demosacing algorithm private, therefore neither Alice nor Bob has the full knowledge of the demosaicing algorithm associated with any $\gamma \in \Gamma$. However, both Alice and Bob can make use of forensic algorithms $\phi(\cdot)$ to estimate demosaicing traces.

To prepare for her attack, Alice first uses $\phi(\cdot)$ to obtain demosaicing filter estimates $\hat{\theta}^{(\ell)}$ associated with target camera model $\gamma^{(\ell)}$ such that

$$\hat{\theta}^{(\ell)} = \phi(I^{(\ell)}). \tag{1}$$

Next, Alice will take an image $I^{(k)}$ from true camera model $\gamma^{(k)}$ and re-sample it using a pre-determined CFA pattern such as the Bayer pattern. Finally, she will re-demosaic the re-sampled image by convolving it with $\hat{\theta}^{(\ell)}$. As a result, the anti-forensic attack $\alpha : \mathcal{I} \times \Theta \rightarrow \mathcal{I}$ and the corresponding attacked image $\tilde{I}^{(k,\ell)}$ are given by

$$\tilde{I}^{(k,\ell)} = \alpha(I^{(k)}, \hat{\theta}^{(\ell)}) = g(I^{(k)}) * \hat{\theta}^{(\ell)}. \tag{2}$$

where $g(\cdot)$ denotes the CFA sampling operation and $*$ denotes convolution.

While this attack has been known for some time, it is currently the state-of-the-art attack aimed at falsifying demosaicing traces, and is capable of succesfully fooling camera model identification algorithms [13, 14]. As a result, if Bob uses a forensic algorithm that relies on demosaicing traces to identify the model of an attacked image $\tilde{I}^{(k,\ell)}$'s source camera, he will incorrectly assume that the image was taken by camera model $\gamma^{(\ell)}$ instead of $\gamma^{(k)}$.

3. COUNTERMEASURE FOR ANTI-FORENSIC CAMERA MODEL IDENTIFICATION

Most commercial cameras apply sophisticated demosaicing algorithms to better preserve local textures and ensure color consistencies within small regions. This introduces complex pixel relationships and cross-channel color correlations into demosaiced images. When Alice falsifies an image using the attack described in Section 2, she attempts to remove the pixel correlations left by the true camera and impose fake demosaicing traces associated with the target camera. However, this linear attack cannot perfectly reproduce demosaicing traces, especially the complex nonlinear and potentially cross-channel color dependencies of the target camera. Additionally, it is possible that part of the demosaicing information from the true camera still remains in falsified images.

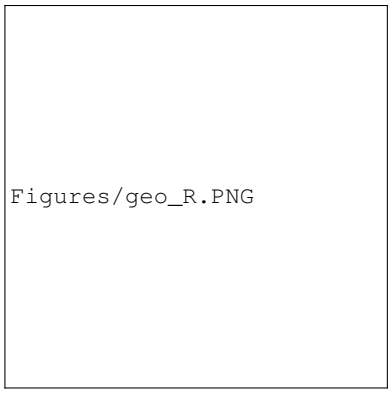
Our framework is designed to detect an anti-forensic attack by suppressing an image's contents and exploiting content-independent color dependencies. A special focus is placed on nonlinear and cross-channel color relationships which an attack has difficulty falsifying and are potentially left by the true camera. To accomplish this, we first re-demosaic the image using a variety of existing demosaicing algorithms as baseline algorithms, then obtain a set of demosaicing residuals by subtracting the re-demosaiced image from the original one. This suppresses image contents and allows us to search for fake traces left by anti-forensics and any remaining traces left by the true source camera in the residuals.

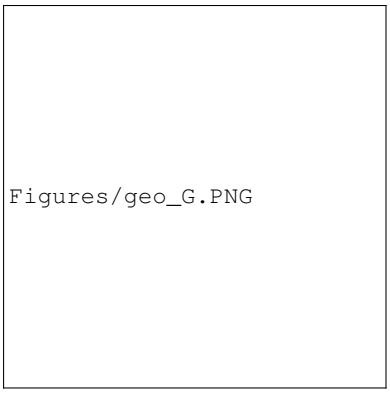
Demosaicing Residuals: To gather each set of demosaicing residuals, we first re-sample the image I using the Bayer pattern to get an image $\tilde{S}$ composed of three partially sampled color channels. Next, we re-demosaic $\tilde{S}$ using a baseline demosaicing algorithm. Let $Demos_H$ be the demosaicing operation associated with demosaicing algorithm H. The set of demosaicing residuals E can be obtained using

$$E = I - Demos_H(\tilde{S}). \tag{3}$$

In our framework, we use seven baseline algorithms to obtain demosaicing residuals and each of them may assist in exposing a distinct aspect of color relationships. We use five linear algorithms including the nearest neighbor, bilinear, bicubic, vertical bilinear and horizontal bilinear algorithms. The last two are modified versions of bilinear algorithm only using directly observed colors in one (vertical or horizontal) direction to interpolate missing values. We also use two nonlinear content-adaptive algorithms alternating projection [20] and local polynomial approximation [21] to uncover more complex pixel dependencies which cannot be faked by the attacker and can tell us valuable information about the true camera models.

Geometric Co-occurrence Patterns: To measure the statistical dependencies among each set of demosaicing residuals, we calculate a set of co-occurrence matrices. Co-occurrence matrices empirically approximate the joint probability distribution of these residuals. It is important, however, to take into account the fact that both a camera





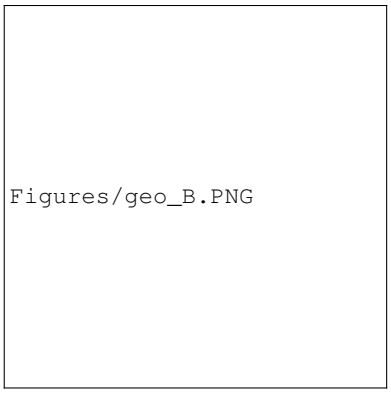
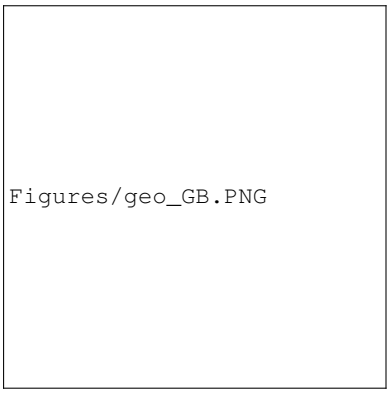


Fig. 1: Intra-channel geometric patterns for red (left), green (middle) and blue (right) channels.



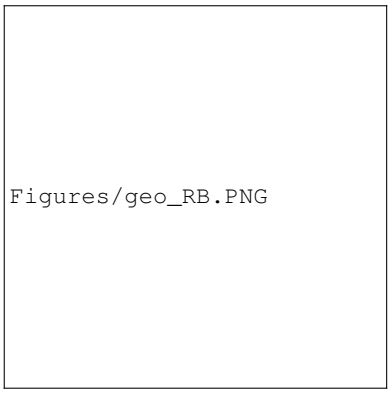


Fig. 2: Inter-channel geometric patterns between red and green (left), green and blue (middle) and red and blue (right) channels.

and an anti-forensic attack interpolate color values with respect to a CFA pattern. Here we propose a new set of co-occurrence patterns to extract from the demosaicing residuals inherent color dependencies aligned with the CFA lattice. Each pattern is designed to capture a specific set of inter-channel or intra-channel dependencies, with particular emphasis placed on the inter-channel (i.e. cross-channel) traces that anti-forensic attacks have difficulty falsifying.

Fig. 1 and Fig. 2 show our 3 intra-channel and 8 inter-channel geometric patterns designed according to the Bayer pattern. Skipping color components directly re-sampled by the CFA pattern, each pattern is specifically designed to capture either intra-channel or inter-channel demosaicing correlation among three color components (d_1, d_2, d_3). We explicitly enforce the diversity among geometric patterns to capture fake color relationships imposed by the attacker and potential remaining demosaicing traces left by the true camera. For patterns 'G', 'RGhv','RB1',RB2' and 'GBhv', we average co-occurrences calculated for several possible color sets to reduce information redundancy caused by location overlapping.

To calculate a co-occurrence matrix for a tuple of color components defined by a geometric pattern, we first discretize demosaicing residuals through quantization with step 2 and truncation with threshold 3. For every repeated CFA lattice throughout the discretized residuals, we extract the set of three color values at locations corresponding to d_1, d_2 and d_3. The co-occurrence matrix is then calculated as the normalized joint-histogram of extracted sets of color values. After calculating all co-occurrence matrices on each geometric pattern from each demosaicing residual, we unite them together as our full feature set for classification.

Falsification Detection Classifiers In reality, Alice and Bob may have different information available to them while launching and detecting an attack. This possible information asymmetry can affect the design and training of our falsification detection classifier. In this paper, we consider two different scenarios. We provide details of how classes are formed and our classifier is trained under each scenario below, followed by a brief description of how our multi-class classifier is constructed.

Table 1: True camera models in our database

Model No.	Camera Model	Model No.	Camera Model
1	Canon PC1234	5	Sansung Galaxy S4
2	Canon Powershot G10	6	iPhone 4s
3	Nikon D7100	7	iPhone 6
4	Samsung Galaxy S3	8	Sony A6000

Scenario I - In this scenario, both Bob and Alice have access to the same set of camera models, i.e. $\Omega = \Gamma$. Here, Bob's goal is to both detect an anti-forensic attack and to identify the true camera model of an attacked image. Under these conditions, our classifier assigns one 'authentic' class for each camera model in Γ along with one 'falsified' class for each unique pair of true and target models in $\Gamma \times \Gamma$ (excluding the pairings of identical true and target models). Bob will create training data for each authentic class by capturing unaltered images using each camera model in Γ, and for each falsified class by launching the attack $\alpha(\cdot)$ for one pairing of true and target camera models. After this, our proposed demosaicing residual co-occurrence features are extracted from each class, and the multi-class classifier described below is trained to distinguish between all possible classes. When examining an image, if the classifier returns an authentic class label, Bob assumes that no attack has be launched. If an image is identified as belonging to a falsified class, Bob knows that the image has been attacked and the image was actually taken by the true camera model associated with that falsified class.

Scenario II - In this scenario, Alice can capture images using at

Table 2: True camera model identification accuracies for 64 classes in our database.

True Model \ Target Model	1	2	3	4	5	6	7	8
1	100.00	100.00	100.00	100.00	99.80	100.00	100.00	100.00
2	100.00	100.00	100.00	99.81	100.00	100.00	99.54	100.00
3	100.00	100.00	100.00	99.90	100.00	100.00	100.00	100.00
4	98.78	100.00	100.00	100.00	100.00	100.00	100.00	100.00
5	99.82	99.60	99.82	100.00	100.00	100.00	99.82	100.00
6	100.00	100.00	100.00	99.79	100.00	100.00	100.00	100.00
7	100.00	100.00	99.59	99.58	99.60	100.00	100.00	99.58
8	100.00	99.58	100.00	100.00	99.62	99.58	100.00	100.00

Table 3: True camera model identification accuracies for 64 classes after JPEG compression.

True Model \ Target Model	1	2	3	4	5	6	7	8
1	100.00	99.80	100.00	100.00	100.00	100.00	100.00	100.00
2	100.00	100.00	100.00	99.81	99.64	99.81	99.69	100.00
3	100.00	99.80	99.90	100.00	100.00	100.00	100.00	99.90
4	100.00	99.39	100.00	100.00	100.00	99.41	100.00	99.41
5	99.82	99.60	100.00	100.00	100.00	99.82	99.82	100.00
6	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
7	100.00	99.14	100.00	99.58	100.00	100.00	100.00	100.00
8	100.00	99.79	100.00	99.79	100.00	100.00	100.00	100.00

Table 4: Negative class testing result of falsified images without access to true camera models.

Identified Class \ True Model No.	6					7					8				
Target Model No.	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Success Fool (%)	*	*	2.98	*	*	*	*	*	*	*	*	*	7.11	*	*
Negative Class (1)	**99.98**	0.41	*	0.10	*	**99.57**	*	0.08	0.04	*	**99.74**	0.25	0.04	0.06	*
Negative Class (2)	0.02	**97.12**	0.08	0.81	0.12	0.34	**99.19**	0.12	0.21	2.61	0.17	**97.56**	0.06	0.74	0.15
Negative Class (3)	*	*	**95.86**	4.58	0.46	*	0.04	**98.84**	0.75	0.20	*	*	**88.75**	3.18	0.06
Negative Class (4)	*	0.90	1.06	**94.27**	0.95	0.09	0.77	0.95	**99.00**	1.54	0.04	2.09	3.94	**95.50**	0.75
Negative Class (5)	*	1.57	0.02	*	**98.74**	*	*	*	*	**95.65**	*	0.84	0.10	0.31	**99.04**

least one camera model that Bob does not have access to, i.e. $\Omega \subset \Gamma$. Here, Bob's goal is simply to detect an attack, since it is impossible for him to accurately identify the true camera model of an attacked image. To detect anti-forensic attacks under this scenario, we propose a strategy we call negative class testing. Our classifier assigns one 'authentic' class and one 'negative' class to each camera model in Ω. Unaltered images from each camera model in Ω are used as training data for each authentic class. To create training data for a camera $\gamma^{(k)}$'s negative class, Bob will use the attack $\alpha(\cdot)$ to falsify images from *all* other true models in Ω with the target model $\gamma^{(k)}$. This is done to provide the classifier with a diverse set of falsified data so that it can detect attacks launched from cameras outside of Ω. After this, our proposed demosaicing residual co-occurrence features are extracted from each class, and the multi-class classifier described below is trained to distinguish between all possible classes. When examining an image, if the classifier returns an authentic class label, Bob assumes that no attack has be launched. If an image is identified as belonging to a negative class, Bob knows that the image has been attacked.

Under both scenarios, our classifier is adapted from the binary ensemble classifier in [22]. We build our multi-class classifier by grouping a set of binary classifiers using the all-vs-all strategy [23]. Specifically, we train a different binary ensemble classifier to distinguish between every possible pair of classes. We then form a multi-class classifier by choosing the class with majority of votes from all binary classifiers as the final decision.

4. EXPERIMENTAL RESULTS

We conducted a series of experiments to fully evaluate our proposed method. We first captured 300 images using each of the 8 camera models in Table 1, and then obtained demosaicing filter estimates associated with each camera model using the improved demosaicing filter estimation method in [7]. Next we applied the anti-forensic attack described in Section 2 to each image to falsify its source camera. Each image was re-demosaiced using the estimated demosaicing filters from each of the other seven camera models and saved as a TIFF. This resulted in 56 falsified classes with unique true and target camera model pairings. When combined with the authentic classes of the 8 camera models, our database consisted of 19,200 images from 64 distinct classes. We divided each image into a set of 512×512 pixel blocks. For the purposes of our experiments, each block is treated as a unique image. Blocks that were too dark or smooth were removed from the database since it is unlikely that real images will consist entirely of dark and smooth regions. Our final dataset consisted of 44,502 blocks from 8 authentic classes and 277,635 blocks from 56 falsified classes.

Attack Detection under Scenario I : In this experiment, we assume Bob has access to all possible camera models. For each block in our database, we first obtained demosaicing residuals using Equation 3, and gathered co-occurrence features using all designed geometric patterns. Then we randomly chose 90% of blocks from each of the 64 classes to train the classifier. Finally, the trained classifier was applied to identify the true camera models of the remaining 10% of blocks. Table 2 shows the percentages of blocks whose true camera models were correctly identified for all 64 classes in our database.

In Table 2, the numbers in the first row and column denote the true and target camera models of each class respectively. Entries on diagonal are identification accuracies for 8 authentic classes. It shows that our framework can successfully detect anti-forensic attacks and identify the true camera models of attacked images with an average accuracy of 99.90%. For 47 out of 64 classes, we achieve 100% accuracy on identifying the true camera models for falsified images. This result demonstrates the ability of our co-occurrence feature set to effectively capture the traces of anti-forensic attack and the remaining demosaicing traces left by the true cameras.

Under Scenario I, we further test the robustness of our method against JPEG compression. We JPEG compressed all blocks in our database with quality factor 90 and repeated feature extraction and classifier training/testing following procedures in the first experiment. Table 3 shows the percentages of JPEG-compressed blocks whose true camera models are correctly identified. The average accuracy for all 64 classes after JPEG compression is 99.91%. This clearly shows that JPEG post-compression does not influence the performance of our method.

Attack Detection under Scenario II: To simulate Scenario II, we assume Bob only has access to cameras in $\Omega = \{1, 2, 3, 4, 5\}$, while Alice may falsify images taken by camera 6, 7, 8 and make them look as if they are taken by cameras in Ω. To create one 'negative class' for each camera in Ω, we used demosaicing estimates associated with it to falsify all images taken by the remaining four cameras . The multi-class ensemble classifier was then trained on five 'negative classes' and five authentic classes. Finally, we applied the

trained classifier to detect anti-forensic attack launched by Alice on the other three unseen camera models.

Table 4 shows our results for anti-forensic attack detection under Scenario II. Numbers in the first and second rows are the true and corresponding target camera model numbers of image blocks attacked by Alice. The third row shows the percentages of image blocks on which our classifier was successfully fooled by the attacker (i.e. identify falsified images as authentically captured by target models). Numbers in the remaining four rows are percentages of blocks identified as negative classes (bracketed number after 'Negative Class' is the corresponding target camera model number of the negative class). For all 15 falsified camera classes, an average of 99.29% image blocks were identified as negative classes. That is, even though we only know the target camera models the attacker intends to present, we can still successfully detect camera model falsification by conducting the negative class testing.

5. CONCLUSION

In this paper, we proposed a new algorithm to detect anti-forensic camera model falsification. Our algorithm operates by characterizing the different content-independent local pixel relationships that are introduced by both authentic demosaicing algorithms and anti-forensic attacks. Our feature set is gathered using both diverse demosaicing residuals, and newly designed CFA-aware geometric patterns. In addition, based on different amounts of knowledge available to the forensic investigator, we proposed two new protocols for constructing and training an ensemble classifier to detect attacks using our feature set. Experimental results show that our algorithm can detect an anti-forensic attack with over 99% accuracy, is robust to JPEG compression, and can even identify the true source camera model in certain circumstances.

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

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

[4] A. Swaminathan, M. Wu, and K. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transaction onInformation Forensics and Security*, vol. 2, no. 1, pp. 91–106, 2007.

[5] 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, Dec. 2009.

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

[7] X. Zhao and M. C. Stamm, "Computationally efficient demosaicing filter estimation for forensic camera model identification," in *International Conference on Image Processing*. IEEE, 2016, pp. 151–155.

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

[9] L. Bondi, L. Baroffio, D. Guera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps towards camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, 2016.

[10] 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.

[11] 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.

[12] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3. IEEE, 2005, pp. III–69.

[13] W.-H. Chuang and M. Wu, "Robustness of color interpolation identification against anti-forensic operations," in *International Workshop on Information Hiding*. Springer, 2012, pp. 16–30.

[14] M. Kirchner and R. Böhme, "Synthesis of color filter array pattern in digital images," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 72 540K–72 540K.

[15] M. C. Stamm, W. S. Lin, and K. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1315–1329, 2012.

[16] S. Lai and R. Böhme, "Countering counter-forensics: the case of jpeg compression," in *Proceedings of the 13th international conference on Information hiding*. Springer-Verlag, 2011, pp. 285–298.

[17] M. C. Stamm, W. S. Lin, and K. R. Liu, "Forensics vs. anti-forensics: A decision and game theoretic framework," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2012, pp. 1749–1752.

[18] M. Barni, Z. Chen, and B. Tondi, "Adversary-aware, data-driven detection of double jpeg compression: How to make counter-forensics harder," in *International Workshop on Information Forensics and Security*. IEEE, Dec 2016, pp. 1–6.

[19] A. Peng, H. Zeng, X. Lin, and X. Kang, "Countering anti-forensics of image resampling," in *International Conference on Image Processing*. IEEE, Sept 2015, pp. 3595–3599.

[20] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE transactions on image processing*, vol. 11, no. 9, pp. 997–1013, 2002.

[21] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy data," *International Journal of Imaging Systems and Technology*, vol. 17, no. 3, pp. 105–122, 2007.

[22] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[23] M. Aly, "Survey on multiclass classification methods," *Neural Netw*, pp. 1–9, 2005.