IMAGE FILTER IDENTIFICATION USING DEMOSAICING RESIDUAL FEATURES

Chen Chen and Matthew C. Stamm

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

ABSTRACT

Image filters have become a popular feature of photo editing software and camera phones. Filter identification can provide useful information for us to determine source and processing history of images. Currently, there is no forensic work done to perform filter identification. In this paper, we propose a framework to search for color correlations left by different filters in a set of interpolation residuals obtained from various demosaicing algorithms. To effectively capture the structures of color correlations, we design a diverse set of geometric co-occurrence patterns and gather both intra-channel and inter-channel color dependencies using co-occurrence matrices. Experiments conducted on two large image databases full demonstrate the ability of our framework to identify a wide range of filters provided by both cameras and third-party software.

Index Terms— Filter identification, Multi-media forensics, Co-occurrence matrix, Ensemble classifier

1. INTRODUCTION

Over the past several years, software applications known as *image filters* have risen in popularity. These applications are typically designed to mimic the effects of optical filters that can be physically attached to a camera's lens. Additionally, some image filters are designed to perform artistic enhancements such as making an image appear as if it was taken by a vintage camera. These filters, which we distinguish from the linear shift-invariant filters frequently discussed in the image processing community, are very widely used to enhance or alter images. Popular social networking applications such as Instagram and Snapchat have image filters integrated to their services. Furthermore, many smartphones include image filtering software into their cameras and third-party image filtering applications are widely available on the Internet.

Since digital images often serve as evidence in many scenarios, it is critical to determine an image's authenticity and processing history before its contents can be trusted. Several forensic algorithms have been developed to detect evidence of image editing and determine an image's origin [1]. The widespread use of image filters, however, may significantly affect the performance of these forensic algorithms. For example, image filters may alter or interfere with traces left by other image manipulations [2, 3]. They may also change camera specific traces used in camera model identification techniques. Furthermore, it is possible that these image filters may effect the performance of steganalysis tools.

Currently, there are no forensic techniques proposed to perform image filter identification. This is a difficult problem because image filters are typically a composite of multiple processing operations. While many forensic algorithms have been proposed to detect individual editing operations that may make up part of an image filter such as resizing and re-sampling [4, 5], contrast enhancement [6], median filtering [7, 8], sharpening [9], and seam carving [10], operations used to make a filter can vary dramatically from filter-to-filter. As a result, none of these techniques can be used to reliably identify all image filters.

While the operations used to make up a filter can be complex and propriety, all image filters change the relationships between an image's color channels. Furthermore, these filters may alter local content-independent pixel relationships within a single color channel. By characterizing these relationships, we can potentially both detect the use of filters and identify the specific filter used to modify an image. Since filters impose color relationships conditioned on the latent color filter array (CFA) structure which is created during a camera's demosaicing process, it is important to take these inherent structures into account when examining the traces of filters.

In this paper, we propose a framework to both determine if an image has been filtered and to identify the specific filter applied. To accomplish this, our framework searches for content-independent pixel value relationships and cross-channel color correlations introduced by image filters. To expose these relationships with respect to the CFA structure, we suppress an image's contents by re-interpolating it using a set of existing demosaicing algorithms, then obtaining interpolation residuals as the difference between each re-interpolated version and the original image. We then characterize statistical dependencies between these residuals using a set of newly design CFA-aware co-occurrence patterns. We use the resulting co-occurrence matrices as a set of features to perform filter detection and identification. Experiments conducted on images modified using both a camera's built-in image filters as well as third-party software show that our proposed approach can identify a wide range of filters with an average accuracy over 97% and is robust to JPEG post-compression.

2. PROBLEM FORMULATION

When performing filter identification, we assume that the filtered images are originally captured by a camera and are filtered only once. The filtered images may be produced directly by a camera after image formation using its built-in software filters. Alternatively, a user can employ third-party image editing software to filter images directly output by a camera. Under both scenarios, the filtered images may possibly be re-compressed by either the camera or the image editing software.

To create their desired effects, most filters provided by current cameras and image-editing software actually perform a composition of multiple processing operations. These operations often adjust the color balance, change the saturation level and apply other perceptual enhancement to photos. Depending on their targeted effects, the internal operations within filters and the specific implementation of

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.

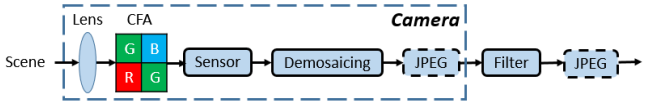

Fig. 1: Possible processing pipeline of a filtered image within and outside a camera.

each operation can be highly complex and vary from filter-to-filter. The implementation details of most filters are usually not publicly available due to commercial reasons. As a result, it is very difficult to build parametric models capable of characterizing all filters. No matter what operations are used to make filters, however, they inevitably leave traces in an image in the form of content-independent color correlations and pixel value dependencies. These can serve as universal traces to identify different filters.

Cameras can also introduce color or pixel value dependencies during the image formation process, primarily during demosaicing. As is shown in Fig. 1, when capturing an image, light reflected from a scene passes through the camera's lens and color filter array (CFA) before hitting the sensor. Due to the CFA's filtering, only one color component is recorded by the sensor at each pixel location. Afterwards, a processing known as demosaicing is responsible for generating color images by interpolating the missing two colors for each pixel from directly recorded ones. This will introduce a set of color dependencies related to the CFA structure. The demosaiced image may then be JPEG compressed by the camera before a filter is applied. We can see that color correlations and pixel value dependences in a filtered image are caused by both cameras' internal demosaicing process and filtering process. As a result, it is important to take CFA structure into account when characterizing filter traces.

3. FILTER IDENTIFICATION FRAMEWORK

To mimic the effects of physical filters attached to the lens of camera, image filters often perform complex operations to artistically adjust or perceptually enhance the colors in images. This not only modifies local pixel value dependencies but also changes the relative color relationships in images. Since the algorithmic implementation of filters can be very complex and highly diverse, finding an effective way to extract general intra-channel and inter-channel color correlations from filtered images becomes a key issue for universal filter identification. Additionally, as described in Section 2, when interpolating missing color components, the demosaicing process within cameras can potentially introduce certain sets of color correlations aligned with CFA pattern into cameras' output images. Therefore, the color relationships in filtered images are actually imposed by filters conditioned on the latent CFA structure.

To better expose color correlations with respect to the CFA structure, we propose a filter identification framework that uses interpolation residuals from demosaicing algorithms to measure color relationships in images according to a CFA pattern. Fig. 3 shows an overview of our proposed framework. We first re-sample color components from images according to a certain CFA pattern, then re-interpolate the missing colors using several demosaicing algorithms. The interpolation residuals are then calculated by subtracting the original filtered images from the re-interpolated ones. This re-interpolation process can suppress image contents and allow the discovery of content-independent color relationships in the interpolation residuals. To capture inherent color correlations exposed in interpolation residuals, we then quantize and truncate residuals and extract a diverse set of CFA-aware co-occurrence features. Finally a multi-class ensemble classifier is applied to utilize color correlation information for filter identification. We now explain the detailed implementation of our framework.

3.1. Re-sampling and Re-interpolation

Before re-interpolating images using different demosaicing algorithms, we re-sample color components according to a predetermined CFA pattern. In our framework, we use the Bayer pattern to sample one color component at each pixel location. Suppose $\mathbf{I}$ is a filtered image under investigation, we denote image after re-sampling as $\tilde{\mathbf{I}}$. Let $Demos_H$ be the re-interpolation operator using demosaicing algorithm H. We then calculate interpolation residual $\mathbf{E}$ as:

$$\mathbf{E} = \mathbf{I} - Demos_H(\tilde{\mathbf{I}}). \quad (1)$$

Since different demosaicing algorithms interpolate color components in different ways, they provide different hypotheses of color correlations in an image. As a result, interpolation residuals obtained from these algorithms should uncover different properties of color correlations in the image. Therefore, we use a diverse set of demosaicing algorithms to measure complex color relationships left by filters. We choose nearest neighbor, bilinear, bicubic, vertical bilinear, horizontal bilinear, alternating projection [11] and local polynomial approximation [12] demosaicing algorithms to perform re-interpolation. The vertical and horizontal bilinear algorithms are two directional versions of the bilinear algorithm which only utilize re-sampled color components in either vertical or horizontal direction for interpolation. Since the chosen seven demosaicing algorithms cover a wide range of demosaicing properties like nonlinearity, cross-channel interpolation and adaptiveness to image contents, they can potentially expose complex and diverse color relationships imposed by different filters.

3.2. Geometric Pattern and Co-occurrence Matrix

To capture color correlations within interpolation residuals, we employ co-occurrence matrices to measure their statistical dependencies. Co-occurrence matrices are empirical approximations of the joint probability of the residual values. By carefully choosing the occurring patterns of different color sets in interpolation residuals, we can gather different types of intra-channel and inter-channel color correlations left by image filters. In order to extract color dependencies with respect to the CFA structure, we make the geometric patterns used to form co-occurrences also CFA-aware. Fig. 3 and Fig. 4 show the 3 intra-channel and 8 inter-channel geometric patterns we designed according to the Bayer pattern respectively.

We can see each geometric pattern specifically defines the locations of a tuple of colors (d_1, d_2, d_3) within the CFA pattern. The color correlation among the three defined colors is then calculated as a 3-rd order co-occurrence matrix. To calculate a co-occurrence given a geometric pattern, we start by quantizing and truncating the interpolation residuals. This is because a co-occurrence matrix essentially computes a discrete joint-histogram. In this paper, after examining the empirical distribution of interpolation residuals, we choose the quantization step as $q = 2$ and truncation threshold as $T = 3$. We then calculate the frequency of the tuple (d_1, d_2, d_3) occurring in every repeated lattice of the CFA pattern throughout the interpolation residuals. Finally, the co-occurrence matrix for the geometric pattern is obtained by normalizing the frequency of its tuple (d_1, d_2, d_3). Note that for intra-channel pattern 'G' and inter-channel patterns 'RGhv','GBhv','RB1','RB2', we average the co-occurrence matrices for four or two tuples to remove possible redundant information due to the overlapping between tuples.

Each of the 11 geometric patterns defines a unique set of colors to capture a particular type of color dependency. As a result, the

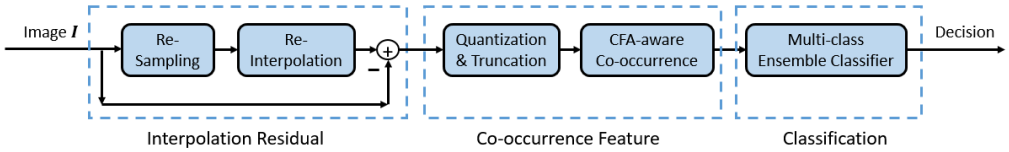

Fig. 2: Full architecture of our proposed filter identification framework.

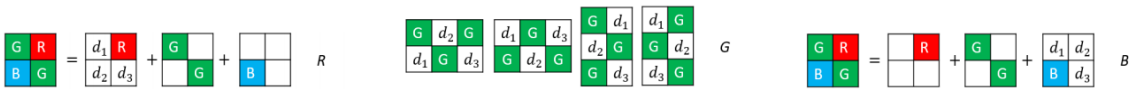

Fig. 3: Intra-channel geometric pattern for red (left), green (middle) and blue (right) channel.

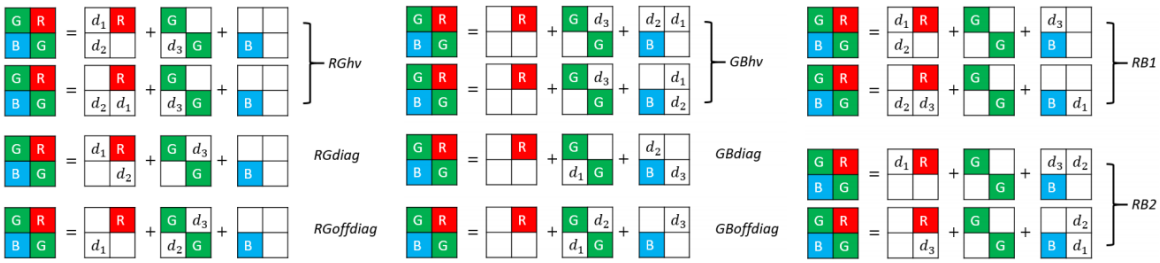

Fig. 4: Inter-channel geometric patterns for red and green (left), green and blue (middle) and red and blue (right) channels.

Table 1: Filter ID and name of built-in filters of iPhone 6S.

Filter ID	Name	Filter ID	Name	Filter ID	Name
1	Chrome	4	Mono	7	Process
2	Fade	5	Noir	8	Tonal
3	Instant	6	None	9	Transfer

color correlations captured by these patterns should be significantly different from each other. Specifically, the intra-channel patterns focus on capturing the statistical correlations of colors introduced by filters performing single-layer enhancement throughout the whole image. The inter-channel geometric patterns focus on uncovering the color layer dependencies caused by filters changing the relative relations of red, green and blue color layers. The diversity among geometric patterns is very critical for collecting a comprehensive set of color correlation information introduced by image filters.

3.3. Multi-class Ensemble Classifier

After calculating co-occurrence matrices using our proposed geometric patterns for each interpolation residual, we unite all co-occurrences together as our full feature set. We then provide these features to a multi-class ensemble classifier to perform filter identification. When training the classifier, it is useful to gather training data for each filter from a variety of camera models. This can help improve the robustness of our method in real scenarios where filters can be applied to images from a wide range of source cameras.

The multi-class ensemble classifier we apply is adapted from the classifier used in [13] based on the binary ensemble classifier proposed in [14]. The binary ensemble classifier is essentially a random forest made from a number of base learners. Each base learner is a Fisher Linear Discriminant (FLD) trained on a random feature subspace and a bootstrap of training samples. The decisions of all FLDs are fused following the majority voting strategy. To construct the multi-class classifier, we train a binary classifier to differentiate every possible pairing of classes. The class receiving majority votes from all binary classifier is then chosen as the decision of the multi-class ensemble classifier.

4. EXPERIMENTS AND RESULTS

4.1. Testing on Camera's Built-in Filters

We conducted a series of experiments to demonstrate the efficacy of our framework. In our first experiment, we tested our framework's ability to determine which filter (if any) was applied to an image while controlling for scene content. We first captured 390 identical scenes using 8 different filters available on an Apple iPhone 6S (iOS 9.2). For each scene, we pre-select the filter we want to add before capturing an image. This ensures that the filters are directly applied by the camera after image formation. Including the directly output images without filtering, our database contained 3,510 images in 9 different classes. Table 1 shows the 9 classes of images in our database. Among the 8 filters provided by iPhone 6S, 'Mono','Noir' and 'Tonal' can only produce grayscale images. For the convenience of description, we named class without filtering as 'None'.

After data collection, we preprocessed all full-size images in our database by first cropping them into 512×512 blocks and then measuring the intensity, texture and flatness of each block using three features proposed in [15]. We only kept blocks with enough illumination and texture for feature extraction, because low quality blocks which are dark, saturated or smooth contain less informative color correlations for classification. This result in a total number of 60,429 image blocks from 9 classes.

After pre-selecting, we extracted co-occurrence features for each selected block according to the framework described in Section 3. 90% of blocks from each class were randomly chosen to train the multi-class ensemble classifier. The remaining 10% of blocks were then tested using the trained classifier. The testing result is shown as a 9×9 confusion matrix in Table 2. Numbers in the first row and column denote the true and predicted filter ID. $*$ means zero percentage and entries highlighted on the diagonal line are correct identification accuracies for each class.

As is shown in Table 2, the average accuracy for 9 classes is 97.56%. For all filters except Filter 4 ('Mono') and Filter 5 ('Noir'), we can achieve over 99% identification accuracies. This result demonstrates that our framework can effectively extract color

Table 2: Confusion matrix of iPhone 6S filters using our desinged feature set.

Predicted Filter \ True Filter	1	2	3	4	5	6	7	8	9
1	99.39	0.12	0.33	*	*	*	*	*	0.50
2	0.12	99.31	*	*	*	0.12	*	*	0.12
3	*	*	99.67	*	*	*	*	*	*
4	*	0.46	*	90.45	8.69	*	*	0.95	*
5	*	*	*	5.39	91.31	*	*	*	*
6	*	*	*	*	*	99.88	*	*	0.37
7	*	*	*	*	*	*	100.00	*	*
8	*	*	*	4.16	*	*	*	99.05	*
9	0.49	0.12	*	*	*	*	*	*	99.00

Table 4: Confusion matrix of filters from Fotor.

Predicted Filter \ True Filter	Bright Spot	Mini Oven	None	Pitts-burgh	Straight Ink
Bright Spot	100.00	0.15	*	*	*
Mini Oven	*	99.70	*	*	*
None	*	*	99.55	*	*
Pittsburgh	*	0.15	*	100.00	*
Straight Ink	*	*	0.45	*	100.00

correlation information from filtered images and accurately detect different filters. We note that for 'Mono' and 'Noir' with relative lower accuracies, most misclassification actually corresponds to confusion between themselves. Fig. 5 shows a comparison between 'Tonal', 'Mono' and 'Noir' of the same scene. We can see that they all produce similar grayscale images, which contain less color correlation information for reliable classification. Especially for 'Mono' and 'Noir', their output images are much darker. This also explains why we want to pre-select informative blocks with relative high quality.

Since rich model for color images (SCRMQ1) [16] has become a state-of-the-art method to universally identify different image manipulations. We conducted a supplementary experiment to compare the performance of our method with SCRMQ1. We extracted the SCRMQ1 feature for all pre-selected blocks in our database and repeated the same training and testing procedure as the previous experiment. The testing result of SCRMQ1 is shown in Table 3.

We can see that the average identification accuracy for SCRMQ1 is 96.00% which is lower than 97.56% achieved by our framework. Additionally, we obtain higher accuracies for almost all image classes except Filter 2 ('Chrome'), for which our accuracy is slightly lower than SCRMQ1 (0.69%). For filters 'Mono' and 'Noir' which are found to be more challenging to differentiate, our method performs much better than SCRMQ1 (roughly 5% improvement). This comparison demonstrates the advantage of our framework over rich model for color images on filter identification. In terms of exposing and extracting color correlations in filtered images, interpolation residuals from demosaicing algorithms and diverse geometric patterns can do a better job than high-pass filters used by rich model.

4.2. Testing on Filters Provided by Third-party Software

To verify that our co-occurrence features don't learn camera model-specific demosaicing information as well as test robustness of our framework against JPEG compression, we built a new database using 6 different camera models. We collected 300 full-size images from Canon PC1234, iPhone 6S, Nikon D7100, Samsung Galaxy Note 4, Samsung Galaxy S5 and Sony A6000. Each model contributed roughly 50 images. We chose an online software Fotor [17] to filter collected images. Fotor provides a large number of filters and

Table 3: Confusion matrix of iPhone 6S filters using rich model for color images (SCRMQ1).

Predicted Filter \ True Filter	1	2	3	4	5	6	7	8	9
1	98.91	*	0.17	*	*	*	0.12	*	1.25
2	*	100.00	*	*	*	0.12	*	*	0.50
3	*	*	98.83	*	*	*	0.12	*	0.25
4	*	*	*	85.80	12.73	*	*	2.86	*
5	*	*	*	10.04	86.29	*	*	*	*
6	*	*	*	*	*	99.75	*	*	0.37
7	*	*	*	*	*	*	99.76	*	0.12
8	*	*	*	4.16	0.98	*	*	97.14	*
9	1.09	*	1.00	*	*	0.12	*	*	97.50

Fig. 5: A comparison between filters 'Tonal' (left), 'Mono' (middle) and 'Noir' (right).

we selected four filters, 'Bright Spot', 'Mini Oven', 'Pittsburgh' and 'Straight Ink', under its 'Classic' category. When filtering images, we chose the 'High' option for JPEG quality. That is, the filtered images were JPEG compressed again. JPEGsnoop showed that the approximate quality factor is 100 which is relative high, but JPEG compression is essentially lossy even with high quality factor. Therefore, we ended up with a new database consisting of 1500 full-size images in 5 classes (plus the unfiltered class) with diverse source cameras and post-JPEG compression applied.

Next, we pre-processed all full-size images of this database in the same way as the first experiment and obtain a total number of 33,275 blocks with acceptable quality. We extracted features for all blocks and trained a multi-class ensemble classifier using 90% of blocks which were randomly chosen. The testing result for remaining 10% of blocks is shown in Table 4.

On the second database, our method achieved an average accuracy of 99.85%. We can successfully detect different image filters with at least 99.55% accuracy. Since the images of this database come from a variety of camera models, this result demonstrates the generalization ability of our method to identify image filters under realistic scenarios. Additionally, since all filtered images were JPEG compressed again after filtering, this result also shows that our method is robust to JPEG compression with relative high quality factors.

5. CONCLUSION

In this paper, we investigate the problem of filter identification and propose a novel framework to effectively capture the color correlation traces left by image filters conditioned on the latent CFA structure. We first expose a diverse set of color correlations by re-interpolating images using various demosaicing algorithms, and design a set of geometric patterns to extract both intra-channel and inter-channel color correlations from interpolation residuals. Experiments on filters from both cameras and third-party software show that our method can accurately identify a wide range of filters with possible post-JPEG compression applied.

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] Matthew C Stamm, Xiaoyu Chu, and KJ Ray Liu, "Forensically determining the order of signal processing operations," in *2013 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2013, pp. 162–167.

[3] Pedro Comesaña and Fernando Pérez-González, "Multimedia operator chain topology and ordering estimation based on detection and information theoretic tools," in *International Workshop on Digital Watermarking*. Springer, 2012, pp. 213–227.

[4] Alin C Popescu and Hany Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on signal processing*, vol. 53, no. 2, pp. 758–767, 2005.

[5] Matthias Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *Proceedings of the 10th ACM workshop on Multimedia and security*. ACM, 2008, pp. 11–20.

[6] Matthew C Stamm and KJ Ray Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 492–506, 2010.

[7] Matthias Kirchner and Jessica Fridrich, "On detection of median filtering in digital images," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, pp. 754110–754110.

[8] Xiangui Kang, Matthew C Stamm, Anjie Peng, and KJ Ray Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1456–1468, 2013.

[9] Gang Cao, Yao Zhao, Rongrong Ni, and Alex C Kot, "Unsharp masking sharpening detection via overshoot artifacts analysis," *IEEE Signal Processing Letters*, vol. 18, no. 10, pp. 603–606, 2011.

[10] Anindya Sarkar, Lakshmanan Nataraj, and Bangalore S Manjunath, "Detection of seam carving and localization of seam insertions in digital images," in *Proceedings of the 11th ACM workshop on Multimedia and security*. ACM, 2009, pp. 107–116.

[11] Bahadir K Gunturk, Yucel Altunbasak, and Russell M Mersereau, "Color plane interpolation using alternating projections," *Image Processing, IEEE Transactions on*, vol. 11, no. 9, pp. 997–1013, 2002.

[12] Dmitriy Paliy, Vladimir Katkovnik, Radu Bilcu, Sakari Alenius, and Karen 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.

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

[14] Jan Kodovskỳ, Jessica Fridrich, and Vojtěch Holub, "Ensemble classifiers for steganalysis of digital media," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 2, pp. 432–444, 2012.

[15] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukáš, "Determining image origin and integrity using sensor noise," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 1, pp. 74–90, 2008.

[16] Miroslav Goljan, Jessica Fridrich, and Rémi Cogranne, "Rich model for steganalysis of color images," in *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*. IEEE, 2014, pp. 185–190.

[17] "Free online photo editor — fotor - photo editing & collage maker & graphic design," `http://www.fotor.com/`, (Accessed on 02/08/2017).