TOWARDS LEARNED COLOR REPRESENTATIONS FOR IMAGE SPLICING DETECTION

Benjamin Hadwiger[⋆]*, Daniele Baracchi*[†]*, Alessandro Piva*[†]*, Christian Riess*[⋆]

[⋆] IT Security Infrastructures Lab, Friedrich-Alexander University Erlangen-Nürnberg
[†] Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze

ABSTRACT

The detection of images that are spliced from multiple sources is one important goal of image forensics. Several methods have been proposed for this task, but particularly since the rise of social media, it is an ongoing challenge to devise forensic approaches that are highly robust to common processing operations such as strong JPEG recompression and downsampling.

In this work, we make a first step towards a novel type of cue for image splicing, which is based on the color formation of an image. We make the assumption that the color formation is a joint result of the camera hardware, the software settings, and the depicted scene, and as such can be used to locate spliced patches that originally stem from different images. To this end, we train a two-stage classifier on the full set of colors from a Macbeth color chart, and compare two patches for their color consistency. Our preliminary results on a challenging dataset on downsampled data of identical scenes indicate that the color distribution can be a useful forensic tool that is highly resistant to JPEG compression.

Index Terms— image forensics, color image formation, splicing detection

1. INTRODUCTION AND RELATED WORK

In passive image forensics, the task is to determine the origin and authenticity of one or more images that do not have an embedded security scheme. Thus, an image has to be validated just from image content and maybe some (limited) additional contextual information about its provenance. Different forensic methods are applicable for this task. The suitability of a particular method oftentimes depends on external factors such as distribution channels and history, the data format, and available contextual information. For a general overview on the field, please refer to one of the books and surveys on the topics, e.g., [1, 2].

Much of the current research activity focuses on statistical image forensics. This family of methods aims at detecting manipulation-induced distortions in the pixel statistics, e.g., by using compression artifacts [3, 4], prior knowledge about resampling [5] or copy-move operations [6], or general noise statistics [7, 8, 9].

A challenge for most forensic methods is to operate with compression and downsampling, which is a commonly used operation in social media. Several physics-based methods that operate on the distribution of light and shadow can be remarkably robust to these operations [10, 11]. However, these methods typically make quite restrictive assumptions on the type of objects that can be analyzed in the scene. Thus, there is an ongoing need for a family of more general methods that are able to cope with strong imaging artifacts.

Fig. 1. Example image from the database by Cheng *et al.* [12] (left), cutout of the Macbeth color chart region (top right), and extracted ground truth in RGB color space (bottom right). In contrast to existing forensic color methods, we not only consider the Macbeth gray patches, but instead represent the full color gamut, i.e., all 24 patches. This allows to characterize the color gamut of the image.

In this work, we explore a new approach to characterize the color image formation in an image. On modern cameras, colors are formed by a combination of camera hardware and software. We hypothesize that the color formation of an image is determined by the hardware platform and the specific software path on the imaging device when the picture is taken. For example, two automatically white-balanced images from the same camera may exhibit quite different color characteristics from one image to another, depending on the way the white balancing is performed. Thus, when an object is spliced from one image into another, and colors are potentially re-balanced, it is still reasonable to assume that the color formation differs between the spliced object and the background.

Towards characterization of the color distribution, we propose a method that learns the color distribution of an image, and how to distinguish two color distributions. To our knowledge, this is the first forensic method that considers the full color gamut instead of a single white point. This is achieved by training a neural network on *all* patches of MacBeth Color chart to model the interplay between the colors of objects and illumination, as illustrated in Fig. 1. We show that this method for determining whether an image patch belongs to the same image is remarkably robust, even to heavy JPEG compression. These initial results are promising, and indicate that color formation can be a promising direction for future forensic research.

Several related works in computer vision tackle the color con-

This material is based on research sponsored by the Air Force Research Laboratory and the Defense Advanced Research Projects Agency under agreement number FA8750-16-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.
The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory and the Defense Advanced Research Projects Agency or the U.S. Government.

stancy problem, i.e., to separate the (intrinsic) body reflection of an object from the spectrum of the light source. Color constancy is oftentimes considered equivalent to the estimation of the light color, by arguing that if the light source is known, the colors in the scene can be factored into illuminant color and object color. To achieve this goal, elementary estimators have been proposed that are either statistical [13, 14], physics-based [15, 16], or learning-based estimators [17, 18].

Selected findings from the fundamental works on color constancy have also been proposed for splicing detection in image forensics. Gholap and Bora [19] proposed to use the intersection of dichromatic planes to detect inconsistencies in specularities. Other works used white point estimators on patches to characterize inconsistencies in the illumination [20, 21]. However, these methods are either somewhat too general for an automated analysis of images, or they have quite restrictive assumptions on the scene composition, e.g., they can only compare pairs of faces [21]. Moreover, these methods are difficult to apply in situations where multiple light sources are present in the scene. Our proposed method differs from these works, as it does not aim to estimate a single white point of the illuminant. Instead, it aims at learning the full color distribution, which, in principle, offers much richer information to model device- or image-specific radiometric distortions.

The paper is organized as follows. The background on color image formation and the method itself are presented in Sec. 2. Experimental results on closed and open image sets are presented in Sec. 3. We conclude the work in Sec. 4.

2. PROPOSED ALGORITHM

The proposed method first learns the color statistics of an image with a Convolutional Neural Network (CNN). Then, a Random Forest determines the consistency of local image patches to expose splices. Such a two-stage approach has recently become popular in statistical image forensics works, e.g., [8, 9, 22].

2.1. Preliminaries

When creating a spliced image from two sources, it is likely that the source images were recorded under different illuminant conditions or with devices with different internal color processing. Thus, our basic assumption is that the color statistics in an original image are consistent, whereas in a spliced image they are not.

We use scenes containing a Macbeth color chart (MCC) for measuring the color distribution, as shown in Fig. 1 (left). This chart contains n_π patches with defined spectral reflectances $\pi_i(\lambda)$, $1 \leq i \leq n_\pi$, where λ represents the wavelength. The color patches appear differently from image to image, depending on the spectral illumination conditions $e(\lambda)$, the vector-valued function $\boldsymbol{c}(\lambda)$ of spectral camera sensitivities for the red, green and blue channel, as well as in-camera color processing operations such as white-balancing, color space transformations and tone mapping. We summarize these operations by the potentially nonlinear function $\boldsymbol{\Omega}$. Thus, an observed image color $\boldsymbol{\Pi}_i$ in an MCC patch i can be modeled as

$$\boldsymbol{\Pi}_i = \boldsymbol{\Omega}\left(\int_\Lambda \pi_i(\lambda) e(\lambda) \boldsymbol{c}(\lambda)\,\mathrm{d}\lambda\right), \tag{1}$$

where the integral is evaluated over the visible spectrum Λ.

Contrary to the forensic scenario, where images *after* the in-camera color processing stack $\boldsymbol{\Omega}$ are analyzed, color constancy algorithms have the goal to estimate and correct an image's color cast *within* the camera pipeline. This color correction is typically applied prior to other potentially nonlinear in-camera color processing operations, such that $\boldsymbol{\Omega}$ can be assumed to be the identity mapping for this task. Thus, color constancy methods regress the observed color $\boldsymbol{\Pi}_w$ of the MCC white patch with $\pi_w(\lambda) \equiv 1$ as single target, which shows the illuminant color filtered by the camera sensitivity:

$$\boldsymbol{\Pi}_w \underset{\boldsymbol{\Omega}\equiv Id}{=} \int_\Lambda e(\lambda)\boldsymbol{c}(\lambda)\,\mathrm{d}\lambda\,. \tag{2}$$

For application in forensics, however, we argue that in order to characterize the illuminant conditions $e(\lambda)$ *and* the full camera color processing pipeline $(\boldsymbol{\Omega}, \boldsymbol{c})$, the MCC white patch alone does not suffice. Instead, we propose to estimate the colors of all color patches in the MCC to characterize the spectral statistics of the imaging conditions and the nonlinear imaging system.

2.2. CNN-based Color Statistics Estimator

To estimate this color statistics, we train a CNN on a set $\mathcal{I}_{tr}^{MCC}$ of images containing MCCs. For each training image $I \in \mathcal{I}_{tr}^{MCC}$, the full set of observed colors $\{\boldsymbol{\Pi}_i^I\}_{i=1}^{n_\pi}$ of the color patches in the MCC is extracted as ground truth, see Figure 1 (bottom right). After removal of the MCC area (Figure 1 top right), each training image I is split into n_p^I patches $\boldsymbol{x}_j^I$, $1 \leq j \leq n_p^I$, of size $M \times N \times 3$.

As done in several color constancy works, e.g. [17, 18], we neglect the luminance values of the ground truth colors and only estimate their chrominance. To this end, the ground truth is transformed to the normalized *r-g*-chromaticity space with

$$\Pi_{i,c}^I = \frac{\Pi_{i,C}^I}{\Pi_{i,R}^I + \Pi_{i,G}^I + \Pi_{i,B}^I}\,, \tag{3}$$

where $C \in \{R, G, B\}$, $c \in \{r, g, b\}$, and $\Pi_{i,C}^I$ and $\Pi_{i,c}^I$ represent a channel of color patch $\boldsymbol{\Pi}_i^I$ in RGB and normalized chromaticity space, respectively. As the three normalized color coordinates $\Pi_{i,c}^I$ sum to 1, it suffices to consider two of them, $\Pi_{i,r}^I$ and $\Pi_{i,g}^I$. This results in a regression target $\boldsymbol{\Pi}^I = \left(\boldsymbol{\Pi}_1^I, \ldots, \boldsymbol{\Pi}_{n_\pi}^I\right) \in [0,1]^{2n_\pi}$ for each image $I \in \mathcal{I}_{tr}^{MCC}$.

The CNN is trained to compute an estimate $\hat{\boldsymbol{\Pi}}_j^I$ of $\boldsymbol{\Pi}^I$ for each $\boldsymbol{x}_j^I$, i.e., it learns a function

$$\hat{\boldsymbol{\Pi}}_j^I = f(\boldsymbol{x}_j^I; \boldsymbol{\theta})\,, \tag{4}$$

with the vector $\boldsymbol{\theta}$ collecting all trainable parameters of the CNN. The objective function uses the mean squared error loss, i.e.,

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{I \in \mathcal{I}_{tr}^{MCC}} \sum_{j=1}^{n_p^I} \left(\boldsymbol{\Pi}^I - f(\boldsymbol{x}_j^I; \boldsymbol{\theta})\right)^2 \tag{5}$$

We base our CNN on the XceptionNet [23] architecture with weights pretrained on ImageNet [24]. The CNN input are patches of size $128 \times 128 \times 3$. We remove the final classification layer of the pretrained XceptionNet and apply global average pooling to the features maps of the previous layer, yielding 2048-dimensional feature maps. Then, a further dense hidden layer with 128 Rectified Linear Units [25], and a dense output layer with $2n_\pi = 48$ units and linear activations are stacked on top.

2.3. Image-level Consistency Assessment

We use the outputs of the trained CNN to characterize the imaging conditions of an image. The consistency of these features extracted from a reference region $\mathcal{R}$ and a test region $\mathcal{T}$ of a test image is then assessed to decide whether it is spliced.

One simple consistency measure is the Euclidian distance between feature vectors. However, to obtain a more flexible decision boundary, we learn the similarity of pairs of CNN estimates $\{\hat{\mathbf{\Pi}}_{j_1}, \hat{\mathbf{\Pi}}_{j_2}\}$ for patch pairs $\{\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2}\}$ from a disjoint set of images $\mathcal{I}_{tr}$. For this, $\hat{\mathbf{\Pi}}_{j_1}$ and $\hat{\mathbf{\Pi}}_{j_2}$ are combined symmetrically with

$$\hat{\mathbf{\Pi}}_{j_1 j_2} = \left(|\hat{\mathbf{\Pi}}_{j_1} - \hat{\mathbf{\Pi}}_{j_2}|, \frac{1}{2}\left(\hat{\mathbf{\Pi}}_{j_1} + \hat{\mathbf{\Pi}}_{j_2}\right)\right), \tag{6}$$

where the absolute value $|\cdot|$ is taken per element. We then train a 2-class Random Forest classifier [26] h with parameters $\boldsymbol{\zeta}$ on these combined feature vectors $\hat{\mathbf{\Pi}}_{j_1 j_2}$ such that

$$h(\hat{\mathbf{\Pi}}_{j_1 j_2}; \boldsymbol{\zeta}) = \begin{cases} 0 & \text{for} \quad \boldsymbol{x}_{j_1} \in I_1, \boldsymbol{x}_{j_2} \in I_2, I_1 = I_2 \\ 1 & \text{for} \quad \boldsymbol{x}_{j_1} \in I_1, \boldsymbol{x}_{j_2} \in I_2, I_1 \neq I_2 \end{cases} \tag{7}$$

For a test image $I \in \mathcal{I}_{te}$, we extract *all* pairs of patches $\{\boldsymbol{x}^I_{j_1}, \boldsymbol{x}^I_{j_2}\}$ with $\boldsymbol{x}^I_{j_1} \in \mathcal{R}$, $\boldsymbol{x}^I_{j_2} \in \mathcal{T}$, $\mathcal{R} \cup \mathcal{T} = I$ and classify the combined feature vectors $\hat{\mathbf{\Pi}}_{j_1 j_2}$ using h. The overall fraction of such pairs classified as inconsistent yields the probability for an image to be spliced.

2.4. Data Augmentation

During training the CNN, the data is augmented with geometric transformations to increase the data diversity and with quality degradation operations to enforce robustness. All transformation parameters are uniformly sampled from their respective value ranges.

The former include rotation and shearing with angles in the range $[-30°, 30°]$, zoom with factors in $[0.8, 1.2]$, and horizontal flipping with probability 0.5. The latter comprise JPEG recompression with a probability of 0.5 and qualities q in the range $[q_{min}, 100]$, followed by JPEG grid desynchronization via horizontal and vertical shifts in the range of $\{-7, \ldots, 7\}$ pixels. Also with probability 0.5, white Gaussian noise with $\sigma = 0.02$ is added.

For training the Random Forest Classifier h, we only apply JPEG compression and desynchronization for data augmentation. With probability 0.5, both patches $\boldsymbol{x}_{j_1}$ and $\boldsymbol{x}_{j_2}$ of each pair are compressed with the same randomly selected level q prior to feature extraction $f(\cdot; \boldsymbol{\theta})$.

3. RESULTS

In this section, we first describe the used datasets, then the performed experiments, and then we discuss the results.

3.1. Datasets

For training the CNN, we use the two datasets by Gehler *et al.* [27] and by Cheng *et al.* [12]. These datasets contain MCCs in each image, as shown in Figure 1 (left). We use the color-processed images instead of the RAW images, as our method requires that the images have been processed by the camera pipelines. The dataset by Gehler *et al.* [27] consists of 568 indoor and outdoor images from 2 cameras, from which 4 underexposed images are excluded. The dataset by Cheng *et al.* [12] consists of 1853 indoor and outdoor images from 9 different cameras. We use the tool CCFind [28] to transfer the provided MCC coordinates from the RAW images to the cropped JPG images. We exclude 126 images which are either over- or underexposed, or for which the MCC extraction failed.

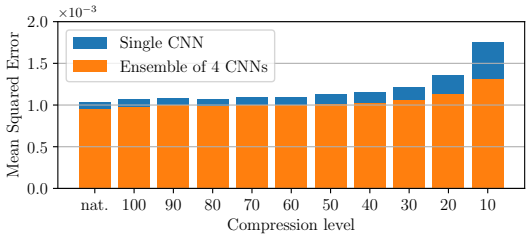

Fig. 2. Mean squared error for clean ("nat.") and JPEG compressed images for single CNN versus average over an ensemble of 4 CNNs.

Both datasets combined yield 2,291 images from 11 cameras. The images are resized with constant aspect ratio to 1,536 pixels in the larger dimension. 10 images per camera are held out for validation and testing. Hence, the CNN is trained on patches from $|\mathcal{I}^{MCC}_{tr}| = 2,071$ images, and the validation and test sets $\mathcal{I}^{MCC}_{val}$ and $\mathcal{I}^{MCC}_{te}$ each comprise 110 images. Patches of 128×128 pixels are extracted with 64 pixels stride for the training and validation images, and with 128 pixels stride for the test images. All patches with MCCs are omitted. Prior to data augmentation, the training, validation and test sets consist of about 630,000, 33,400 and 9,400 patches, respectively.

For training the Random Forest Classifier, we use $|\mathcal{I}_{tr}| = 1,100$ images from 11 different smartphone cameras from the VISION Dataset by Shullani *et al.* [29]. Here, the images are all resized to $1,536 \times 1,024$ pixels prior to patch extraction.

The test images $\mathcal{I}_{te}$ for splicing detection are built from 332 images of 83 scenes of the Dresden Image Database [30] (see Sec. 3.4).

3.2. Training and Evaluation of the CNN

We train the CNN with the Adam optimizer [32] with an initial learning rate $l = 10^{-3}$ and the parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Whenever the loss on the validation data does not improve for 3 epochs, we reduce l by a factor of 10. We applied early stopping once the validation loss does not significantly improve anymore. That way, the model trained for 14 epochs to obtain a final validation MSE loss of $1.018 \cdot 10^{-3}$. Training takes approximately 11 hours on an NVIDIA GeForce GTX 1080 GPU using Python and Keras.

We evaluate the CNN on the MCC test images $\mathcal{I}^{MCC}_{te}$ for different compression levels. In particular, we computed the mean squared error on the clean images, and on the same images with JPEG compressions of qualities 100 to 10 in steps of 10. The results are shown in Fig. 2 for a estimates of a single CNN trained with $q_{min} = 50$, and for median averaged estimates over an ensemble of 4 CNNs, where 2 are trained with $q_{min} = 50$ and 2 with $q_{min} = 10$.

We note that the estimates are very robust with respect to JPEG compression. Also, as each CNN in the ensemble is trained with different randomly sampled batches, the median of their outputs yields more stable estimates. For this reason, unless otherwise stated, we conduct all following experiments with that ensemble of CNNs.

3.3. Closed-Set Patch Association

As a baseline, we perform a closed-set experiment to investigate the descriptiveness of the learned color statistics for the image provenance of a patch. To this end, each test image $I \in \mathcal{I}^{MCC}_{te}$ is subdi-

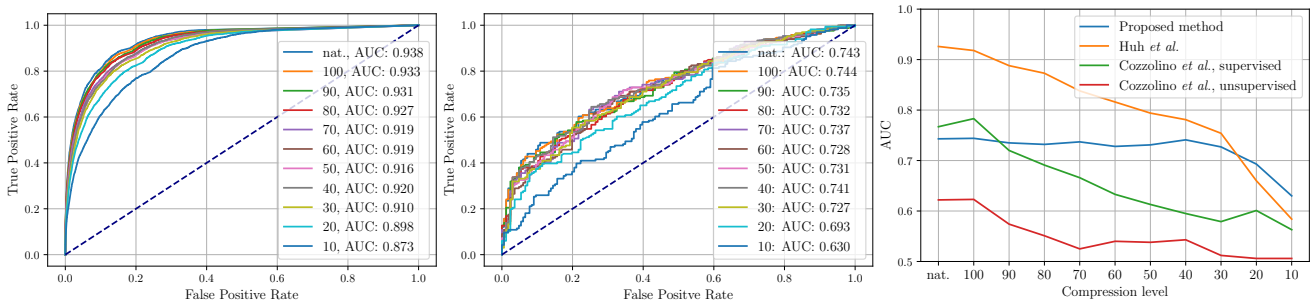

Fig. 3. ROC curves for clean ("nat") and JPEG compressed images. Left: on the Closed Set. Middle: on the Open Set. Right: AUCs of Cozzolino *et al.* [31] unsupervised (red) and supervised (green), Huh *et al.* [22] (orange), and the proposed method (blue) on the Open Set.

vided into non-overlapping patches, and randomly split into training and test set with a ratio of $0.7/0.3$. Then, we train a Random Forest classifier on this training set with the number of classes equal to number of test images $|\mathcal{I}_{te}^{MCC}| = 110$. For evaluation, we classify the remaining test patches. The micro-averaged ROC curves are reported in Figure 3 (left).

For uncompressed images, the AUC is 0.938, indicating that the learned color statistics are indeed characteristic for the image provenance of a patch. Additionally, for JPEG quality 10, the AUC is still 0.87, which indicates that the features are promising to also work on low-quality data.

3.4. Open-Set Splicing Detection

To evaluate the learned color statistics in a forensically more realistic scenario, we created a completely separate test dataset $\mathcal{I}_{te}$ using the Dresden Image Database [30]. For each of the 83 scenes, we randomly selected 4 images from different cameras, resized to $1,024 \times 1,536$ pixels. Two of the images are used as pristine images. From the two remaining images, we insert the left third of the image area into one of the pristine images to obtain two splices. Since the content of the scenes in the Dresden database is aligned, the image content of the splices is highly consistent, which prevents the the classifier to guess splices from varying image content.

We evaluate our method on the test dataset $\mathcal{I}_{te}$ with detection scores obtained as described in Sec. 2.3, using a Random Forest with 500 trees pruned to a maximum depth of 20.

The training data $\mathcal{I}_{tr}$ is created from 1,100 images of 11 smartphone cameras of different brands from the VISION database. We sample patches of 128×128 pixels to obtain per database image 100 pairs from within the image, and 100 pairs from one patch within the image, and the second patch from another image. Pairs are concatenated via Eqn. 6, yielding in total 220,000 combined samples.

Results for this experiment are shown in Fig. 3 (middle). The AUCs range from 0.74 for high quality images, to 0.69 for JPEG quality 20, and 0.63 for JPEG quality 10, which underlines the robustness of the color features to compression. Qualitative examples are shown in Fig. 4. The top row shows spliced images, and the bottom row detection heat maps over the left (spliced) third of the image. From left to right, the first two images exhibit high manipulation probabilities, the second two images show much lower probabilities.

3.5. Comparison to Related Work

Comparisons to the recent work by Huh *et al.* [22] and Cozzolino *et al.* [31] are shown in Fig. 3 (right) for various compression levels.

Fig. 4. Qualitative results on four spliced images (top) with compression 20 and the resulting heat maps on the left third of the image (bottom). The two cases on the left show high splicing probabilities, the two cases on the right are failure cases with low probabilities.

We evaluated the method by Cozzolino *et al.* in unsupervised (red) and supervised (green) mode; in the latter case, the unmodified area of an image is provided as additional training input. In both cases, the proposed method (blue) outperforms the method by Cozzolino *et al.*, which can be attributed to the fact that much of the low-level statistics is removed upon resampling and recompressing the test data. For the method by Huh *et al.* (orange), we aggregate scores to an image-level decision analogously to 2.3. It performs remarkably well. We hypothesize that this is due to the fact that, as the authors state, it learns a mixture of color artifacts, and due to in parts inhomogeneous resampling of the test data (for which our method has no training examples). However, our method outperforms Huh *et al.* for very low JPEG qualities, which again shows the robustness of the proposed color cue.

4. CONCLUSIONS

We demonstrated that the color formation of an image can be used to determine whether an image patch belongs to a source image. One key idea of this approach is to use all patches from a Macbeth color chart to train a specialized network for the extraction of color features, which are subsequently used to compare patches in an image. Our evaluation on resampled data shows that the proposed color cues are highly robust to JPEG compression.

This is preliminary work, with a number of possibilities to follow up. For example, in future work, we plan to incorporate additional prior knowledge on camera image formation into the feature extraction. We also plan to perform the consistency assessment on a manifold in paired feature space using a Siamese network.

5. REFERENCES

[1] H. Farid, *Photo Forensics*, The MIT Press, 2016.

[2] J. Redi, W. Taktak, and J.-L. Dugelay, "Digital Image Forensics: A Booklet for Beginners," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 133–162, Jan. 2011.

[3] T. H. Thai, R. Cogranne, F. Retraint, and T. Doan, "JPEG Quantization Step Estimation and Its Applications to Digital Image Forensics," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 123–133, Jan. 2017.

[4] T. Bianchi, A. Piva, and F. Perez-Gonzalez, "Near Optimal Detection of Quantized Signals and Application to JPEG Forensics," in *IEEE International Workshop on Information Forensics and Security*, Nov. 2013, pp. 168–173.

[5] M. Kirchner, "Linear Row and Column Predictors for the Analysis of Resized Images," in *ACM SIGMM Multimedia & Security Workshop*, Sept. 2010, pp. 13–18.

[6] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An Evaluation of Popular Copy-Move Forgery Detection Approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, June 2012.

[7] J. Fridrich and J. Kodovský, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, June 2012.

[8] O. Mayer and M. C. Stamm, "Learned forensic source similarity for unknown camera models," in *International Conference on Acoustics, Speech and Signal Processing*, Apr. 2018.

[9] D. Cozzolino and L. Verdoliva, "Noiseprint: a CNN-based camera model fingerprint," arxiv preprint, University of Naples, 2018, arXiv:1808.08396.

[10] E. Kee, J. F. O'Brien, and H. Farid, "Exposing Photo Manipulation from Shading and Shadows," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 165:1–165:21, Aug. 2014.

[11] B. Peng, W. Wang, J. D., and T. Tan, "Optimized 3D Lighting Environment Estimation for Image Forgery Detection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 479–494, Feb. 2017.

[12] D. Cheng and D. K. Prasad und M. S. Brown, "Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution," *Journal of the Optical Society of America A*, vol. 31, no. 5, pp. 1049–1058, May 2014.

[13] J. van de Weijer, T. Gevers, and A. Gijsenij, "Edge-Based Color Constancy," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2207–2214, Sept. 2007.

[14] A. Gijsenij, T. Gevers, and J. van de Weijer, "Improving Color Constancy by Photometric Edge Weighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 918–929, May 2012.

[15] R. Tan, K. Nishino, and K. Ikeuchi, "Color Constancy through Inverse-Intensity Chromaticity Space," *Journal of the Optical Society of America A*, vol. 21, no. 3, pp. 321–334, Mar. 2004.

[16] J. Toro and B. Funt, "A Multilinear Constraint on Dichromatic Planes for Illumination Estimation," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 92–97, Jan. 2007.

[17] G. D. Finlayson, S. D. Hordley, and I. Tastl, "Gamut Constrained Illuminant Estimation," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 93–109, Apr. 2006.

[18] J. T. Barron, "Convolutional Color Constancy," in *IEEE International Conference on Computer Vision*, 2015, pp. 379–387.

[19] S. Gholap and P. K. Bora, "Illuminant Colour based Image Forensics," in *Proceedings of the IEEE Region 10 Conference*, 2008, pp. 1–5.

[20] C. Riess and E. Angelopoulou, "Scene Illumination as an Indicator of Image Manipulation," in *International Information Hiding Conference*, 2010, pp. 66–80.

[21] T. J. de Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, , and A. Rocha, "Exposing Digital Image Forgeries by Illumination Color Classification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, July 2013.

[22] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting Fake News: Image Splice Detection via Learned Self-Consistency," Tech. Rep., arXiv:1805.04096v3, Sept. 2018, `https://arxiv.org/pdf/1805.04096.pdf`.

[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 1800–1807.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[25] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

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

[27] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp, "Bayesian Color Constancy Revisited," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.

[28] K. Hirakawa, "Colorchecker finder," `http://issl.udayton.edu/index.php/research/ccfind/`.

[29] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, p. 2017:15, Dec. 2017, `https://doi.org/10.1186/s13635-017-0067-2`.

[30] T. Gloe and R. Böhme, "The'dresden image database'for benchmarking digital image forensics," in *Proceedings of the 2010 ACM Symposium on Applied Computing*. ACM, 2010, pp. 1584–1590.

[31] D. Cozzolino, G. Poggi, and L. Verdoliva, "Splicebuster: A new blind image splicing detector," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Nov. 2015.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.