AUGMENTED CONVOLUTIONAL FEATURE MAPS FOR ROBUST CNN-BASED CAMERA MODEL IDENTIFICATION

Belhassen Bayar and Matthew C. Stamm

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

ABSTRACT

Identifying the model of the camera that captured an image is an important forensic problem. While several algorithms have been proposed to accomplish this, their performance degrades significantly if the image is subject to post-processing. This is problematic since social media applications and photo-sharing websites typically resize and recompress images. In this paper, we propose a new convolutional neural network based approach to performing camera model identification that is robust to resampling and recompression. To accomplish this, we propose a new approach to low-level feature extraction that uses both a constrained convolutional layer and a nonlinear residual feature extractor in parallel. The feature maps produced by both of these layers are then concatenated and passed to subsequent convolutional layers for further feature extraction. Experimental results show that our proposed approach can significantly improve camera model identification performance in resampled and recompressed images.

Index Terms— Camera identification, convolutional neural networks, constrained convolutional layer, deep convolutional features.

1. INTRODUCTION

Digital images are frequently used in important settings, such as evidence in legal proceedings or criminal investigations. In order to trust these images, it is important to verify their source. As a result, many forensic algorithms have been developed to blindly determine the make and model of an image's source camera [1]. Forensic algorithms have been developed to perform camera model identification using a wide variety of features including heuristically designed statistical metrics [2], linear estimates of demosaicing traces [3, 4, 5], demosaicing residuals [6, 7] and rich model features [8].

Recently, researchers have begun adapting convolutional neural networks to perform forensic tasks. Initial CNNs designed to perform camera model identification operate by first suppressing an image's contents and extracting low-level using a fixed high-pass filter [9]. These low-level features are then passed to a CNN. This approach was first proposed in steganalysis research [10]. Alternatively, low-level forensic features can be adaptively learned using a layer known as a constrained convolutional layer [11]. This layer, which was initially proposed to learn image manipulation detection features, is able to jointly suppress an image's content and adaptively learn diverse set of linear prediction residuals while training a CNN.

While these approaches can succesfully identify the model of an unaltered image's source camera, their performance often degrades signficantly if the image is subjected to post-processing. This is especially the case for operations such as resizing/resampling and JPEG compression. This performance degradation is an important problem since online photo-sharing websites and social media applications often resize and recompress an image. In order for camera model identification techniques to work in real world scenarios, it is important to devise methods to make them robust to these common post-processing operations.

Other areas of forensics, such as manipulation detection, have encountered a similar problems with robustness to post-processing. Work in these areas suggests that in some scenarios, utilizing nonlinear residuals can potentially increase an algorithm's robustness to post-processing. [12, 13]. As a result, researchers may ask: Can the addition of low-level nonlinear residuals such as the median filter residual (MFR) [12] increase the robustness of camera model identification algorithms? Since convolutional filters used by CNNs are only able to learn linear feature extractors, how should low-level nonlinear residual features be integrated into a CNN? Should they replace linear feature extractors such as a constrained convolutional layer or a fixed high-pass filter, or should another strategy be adopted?

In this paper, we propose a new CNN-based camera model identification approach that is robust to resampling and JPEG recompression. To accomplish this, we propose a new approach to low-level forensic feature extraction that we call augmented convolutional feature maps (ACFM). In this approach, both a constrained convolutional layer and a nonlinear residual feature extractor such as the MFR are used in parallel. The feature maps produced by both of these layers are then concatenated and higher level associations between these feature maps are learned by subsequent convolutional layers. Through a set of experiments, we evaluate both the accuracy and robustness of our camera model identification approach on a set of images from 26 camera models from the Dresden Image Database [14] subject to both resampling and JPEG post-compression. Experimental results show that our proposed ACFM approach can improve a CNN's camera model identification robustness to resampling and JPEG compression.

2. AUGMENTED CONVOLUTIONAL FEATURE MAPS

Recently, CNN-based forensic approaches have been proposed to individually make use of adaptive linear prediction-error feature extractors [11, 15, 16] as well as fixed nonlinear residual extractors [17]. Both of these feature extractors have their own set of advantages. When a CNN is used with an adaptive feature extractor such as a constrained convolutional layer, it has shown the ability to learn a diverse set of prediction residual features that outperform the fixed linear residuals proposed in [10, 9] for forensic tasks [18]. Al-

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.

ternatively, nonlinear residual features such as the MFR have been experimentally shown to improve the robustness of manipulation detection CNNs to JPEG compression [13].

In order to perform robust camera model identification in post-processed images, we need to retain at the same time the adaptively learned linear prediction residual features produced by a constrained convolutional layer as well as nonlinear residual features such as the MFR. This suggests, we do not want to replace the first layer of a CNN with a fixed nonlinear residual feature extractor. At the same time, we want to integrate these low-level nonlinear residuals into our CNN.

To solve this problem, we propose a new approach to low-level residual feature extraction that we call augmented convolutional feature maps (ACFM). In this approach, a fixed nonlinear residual feature extractor is placed in parallel with a set of constrained convolutional filters. The feature maps produced by the constrained covolutional layer are concatenated with the nonlinear feature residuals to create an augmented set of feature maps. This augmented set of feature maps is then directly passed to a regular convolutional layer. Deeper convolutional layers in the CNN will learn higher-level features and associations between these linear and nonlinear residuals.

During training, the nonlinear feature extractors are held constant, while filters in the constrained convolutional layer are updated through stochastic gradient descent. In this way a diverse set of linear prediction-error feature extractors are learned that compliment the nonlinear residual features which are mainly used to increase robustness of CNNs to post-processing.

In this paper, we form our augmented convolutional feature maps using the nonlinear MFR features [12, 13]. The MFR is formally defined as

$$d(i,j) = x(i,j) - med_3(i,j), \tag{1}$$

where $x(i,j)$ is the $(i,j)^{th}$ pixel in the subject image and $med_3(i,j)$ is the $(i,j)^{th}$ pixel in the median filtered form of the same image using a 3×3 kernel. These nonlinear pre-computed MFR features will take the form of a single feature map then will be used to augment the feature maps produced by the constrained convolutional layer used in parallel.

The constrained convolutional layer, which is used in parallel with the nonlinear residual layer, is designed to adaptively learn feature extractors that take the form of prediction error filters. This is accomplished by actively enforcing the following prediction-error filter constraints

$$\begin{cases} \boldsymbol{w}_k^{(1)}(0,0) = -1, \\ \sum_{m,n\neq 0} \boldsymbol{w}_k^{(1)}(m,n) = 1, \end{cases} \tag{2}$$

during training on each of the K filters $\boldsymbol{w}_k^{(1)}$ in the constrained convolutional layer. In this manner, training proceeds by updating the filter weights $\boldsymbol{w}_k^{(1)}$ at each iteration using the stochastic gradient descent algorithm during the backpropagation step, then projecting the updated filter weights back into the feasible set by reinforcing the constraints in (2). Pseudocode outlining this process is given in Algorithm 1.

Algorithm 1 Training algorithm for constrained convolutional layer

Initilize $\boldsymbol{w}_k$'s using randomly drawn weights
i=1
while $i \leq max\_iter$ **do**
  Do feedforward pass
  Update filter weights through stochastic gradient descent and backpropagate errors
  Set $\boldsymbol{w}_k(0,0)^{(1)} = 0$ for all K filters
  Normalize $\boldsymbol{w}_k^{(1)}$'s such that $\sum_{\ell,m\neq 0} \boldsymbol{w}_k^{(1)}(\ell,m) = 1$
  Set $\boldsymbol{w}_k(0,0)^{(1)} = -1$ for all K filters
  i = i+1
  if training accuracy converges **then**
    exit
end

3. NETWORK ARCHITECTURE

In this section, we give an explicit description about how to implement the feature maps augmentation approach using the nonlinear MFR features. Fig. 1 depicts the overall architecture of our proposed ACFM-based CNN. Our CNN has the ability to : (i) jointly suppress an image's content and adaptively learn low-level linear residual features while training the network, (ii) perform convolutional feature maps augmentation using linear and nonlinear residuals (iii) extract higher-level features through deep layers and (iv) learn new association between higher-level augmented feature maps using 1×1 convolutional filters. These type of filters are used to learn linear combination of features located in different feature maps but located at the same spatial location. In what follows, we give a brief overview about each conceptual block that we used in our architecture.

3.1. Convolutional feature maps augmentation

Because CNNs in their existing form tend to learn features related to an image's content, we make use of a contrained convolutional layer ("Constrained Conv") in our architecture to jointly suppress the image's content and adaptively learn linear residual features. The "Constrained Conv" layer consists of three constrained convolutional filters of size $5\times5\times1\times1$ each which operate with a stride of 1. This layer yields three prediction-error feature maps. The *adaptively learned* linear residuals take the form of low-level pixel-value dependency features.

However, when images are post-compressed, existing linear feature extractors in CNN, such as convolution, may not capture all camera model identification features. Therefore, we propose to augment the "Constrained Conv" layer output feature maps in CNN with the nonlinear MFR by using a concatenation layer called "CFMA" (see Fig. 1).

To accomplish this, the input layer of CNN consists of two-channel image. The first channel corresponds to a 256×256 green layer image while second channel corresponds to the computed MFR features of the same image. In order to concatenate the prediction-error features with MFR, the feature maps output of "Constrained Conv" layer and the MFR features should have same dimension, i.e, 252×252. To do this, the MFR channel of the input layer is first convolved with a fixed 5×5 identity filter with a stride of 1.

Subsequently, the outputs of "Constrained Conv" and "Identity Conv" layers are concatenated using the "CFMA" layer. Thus, the "Constrained Conv" layer's output is augmented from $252\times252\times3$ to $252\times252\times4$. Deeper layers in our proposed CNN will learn new associations between MFR and prediction-error features. In our experiments, when CNN is used without ACFM we use the architecture in red dashed line in Fig. 1 that we call NonACFM-based CNN. Note that our ACFM approach can be generalized and used with other types of nonlinear features to introduce diversity to the existing features and increase the robustness of CNN in real world

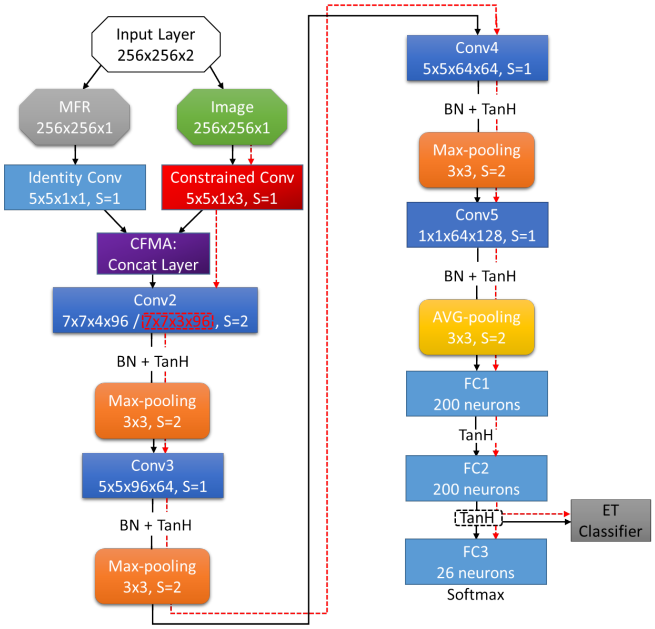

Fig. 1: Our proposed ACFM-based CNN architectures; red dashed line: NonACFM-based CNN; CFMA: Convolutional Feature Maps Augmentation; BN:Batch-Normalization Layer; TanH: Hyperbolic Tangent Layer; ET: Extremely Randomized Trees

scenarios.

3.2. Hierarchical feature extraction

To learn higher-level classification features, the low-level prediction-error feature maps augmented by MFR features are directly passed to a sequence of three regular convolutional layers followed by one 1×1 convolutional layer. From Fig. 1 we can see that every convolutional layer in our CNN is followed by a batch normalization layer (BN) [19], a hyperbolic tangent (TanH) activation function then a pooling layer. By contrast, the "Constrained Conv" and "Indentity Conv" layers are not followed by any type of nonlinear operation since residual features can be destroyed by any of these types of nonlinear operations. Additionally, we use the max-pooling layer after all the regular convolutional layers, whereas, an average-pooling layer is used after the 1×1 convolutional layer to preserve the most representative features. We use these filters to learn new association across the highest-level convolutional feature maps in our CNN. The size of each convolutional filter as well as the stride size are reported in Fig. 1. Note that when CNN is used without feature maps augmentation (i.e., NonACFM-based CNN) the "Conv2" layer is of dimension $7 \times 7 \times 3 \times 96$.

3.3. Classification

The output of the hierarchical convolutional layers is fed to a regular neural network which consists of three fully-connected layers to perform classification. The two first layers have 200 neurons and followed by a TanH activation function, whereas, the output layer is followed by a softmax activation function. The number of neurons in the output layer is equal to the number of the considered camera models, i.e. 26 camera models from the Dresden Image Database [14].

In different applications, other classifiers have shown to perform better than a softmax-based classifier. Therefore, in order to improve CNN's peroformance we adapt an alternate classification strategy using the "deep features" approach [20]. To accomplish this, we use the activation levels of each neuron in the second fully connected layer as a set of deep features. These are then passed to an extremely randomized trees (ET) classifier [21] to perform camera model identification on the basis of these features.

4. EXPERIMENTS

We conducted a set of experiments to evaluate the effectiveness and robustness of our proposed ACFM-based CNN to perform camera model identification. To study the impact of augmenting a CNN with nonlinear MFR features, we compared our CNN to an architecture which did not include feature map augmentation, i.e., a NonACFM-based CNN that used only a "Constrained Conv" layer to extract low-level linear residual features. We then compared our results to a CNN using a fixed high-pass filter to perform low-level feature extraction as proposed in [9], i.e., HPF-based CNN.

To build our experimental dataset we downloaded images from the publicly available Dresden Image Database [14]. We collected $15,000$ images for the training and testing data which consists of 26 camera models. To train our CNNs we randomly selected $12,000$ images from our experimental database. Next, we divided these images into 256×256 pixel patches and retained all the 36 central patches from the green layer of each image for our training database. Each patch corresponds to a new image which is associated with one camera model class. In total, our training database consisted of $432,000$ patches.

When training each CNN, we set the batch size equal to 64 and the parameters of the stochastic gradient descent as follows: $momentum = 0.9$, $decay = 0.0005$, and a learning rate $\epsilon = 10^{-3}$ that decreases every 4 epochs ($27,000$ iterations) by a factor $\gamma = 0.5$. We trained the CNN in each experiment for 44 epochs ($297,000$ iterations).

To evaluate the performance of our proposed approach, we created a testing database by dividing the $3,000$ images not used for the training into 256×256 pixel patches in the same manner described above. In total, our testing database consisted of $108,000$ patches. We then used our CNN to identify the source camera model of each image in the testing set. Note that training and testing are disjoint.

When digital images are downloaded or uploaded, they are commonly resampled and/or JPEG compressed. In order to mimic real world scenarios, we created seven corresponding experimental databases for the training and testing patches by applying the following image editing operations to our unaltered collected database: JPEG compression (QF=90), resampling (i.e., downscaling by 90% and 50% as well as upscaling by 120%) and resampling followed by a JPEG post-compression (QF=90). Furthermore, with non-resampled patches we use 256×256 input layer in our CNN, whereas, with resampled patches the size of the input layer is set according to the rescaling factor, i.e., 307×307 with 120% upscaling, 230×230 with 90% downscaling and 128×128 with 50% downscaling.

4.1. ACFM-based CNN with median filter residual

We assessed the robustness of our approach using the augmented convolutional feature maps (ACFM)-based CNN in Fig. 1 then we compared it to a NonACFM-based CNN (red dashed line) where MFR features are not introduced to the network. To do this, for each

Table 1: CNN's identification rate on processed images using Softmax layer (top) and Extremely Randomized Trees (ET) classifier (bottom).

	Resampling			**Resampling + JPEG (QF=90)**			**JPEG**	**Original**
Methods	**120%**	**90%**	**50%**	**120%**	**90%**	**50%**	**QF = 90**	–
ACFM-based CNN	**97.14%**	**95.93%**	**90.75%**	93.86%	91.42%	**79.31%**	**97.26%**	**98.26%**
NonACFM-based CNN	96.75%	95.76%	87.70%	**94.94%**	**91.89%**	75.68%	97.23%	98.24%
HPF-based CNN	95.94%	95.68%	87.54%	90.45%	83.96%	67.16%	96.00%	97.52%
Top: Softmax-based CNN; Bottom: ET-based CNN								
ACFM-based CNN	**97.61%**	**96.44%**	**91.47%**	94.71%	92.10%	**79.74%**	**97.63%**	**98.58%**
NonACFM-based CNN	97.28%	96.38%	88.88%	**95.50%**	**92.50%**	76.05%	97.60%	98.52%
HPF-based CNN	96.47%	96.14%	88.67%	91.31%	84.71%	67.42%	96.36%	97.83%

experimental database (training and testing data) we computed the nonlinear MFR of each patch then added it to the existing patch as a second channel. That is, the CNN's input $256\times256\times1$ patches for instance become of dimension $256\times256\times2$. We trained and tested our ACFM-based CNN as described above. Table 1 summarizes the results of all our experiments using both a softmax classification layer and an Extremely Randomized Trees (ET) classifier as described in Section 3.3. We compare our proposed augmented feature maps-based approach to its homologue NonACFM-based CNN that only uses linear residuals learned through a "Constrained Conv" layer.

Our experimental results show that augmenting feature maps produced by the constrained convolutional layer with MFR features can typically improve the overall identification rate with all possible tampering operations. Noticeably, it can achieve 98.58% identification rate with unaltered images and at least 79.74% with 50% downscaled then post-compressed (QF=90) images using an ET classifier. From Table 1, one can observe that the ET-based CNN approach outperforms the softmax-based CNN.

When we use the augmented feature maps-based CNN, one can notice that with 50% downscaling, we can improve the camera model identification rate over an architecture which did not use nonlinear MFR features by 3.69% with JPEG post-compression and by 2.59% without JPEG post-compression (see Table 1). This demonstrates that introducing the nonlinear MFR features to the network can improve the robustness of CNN against real world scenario processing operations.

Finally, we would like to make sure that our proposed ACFM-based CNN did not learn higher-level features only related to MFR features. That is, learning the association between prediction-error features and MFR throughout deeper layers improves CNN's performance. To accomplish this, we trained the NonACFM-based CNN architecture without a "Constrained Conv" layer (i.e., without prediction-error features) and using the MFR as an input layer to CNN. We used our original experimental database as well as the 50% downscaling with and without JPEG post-compression databases where the augmented feature maps-based approach significantly outperforms its homologue. Our experiments show that ACFM-based CNN and NonACFM-based CNN both outperform CNN that only used MFR features. The latter MFR-based CNN can only achieve an identification rate equal to 96.60% with unaltered images, 62.50% with 50% downscaled then JPEG post-compressed images and 74.99% with 50% downscaled database. This experimentally demonstrates that learning the association between prediction-error features and MFR throughout deeper layers in the network significantly increases the robustness of CNN in real world scenarios.

4.2. Comparison with non-adaptive linear residual extractors

As mentioned above, early approaches to perform CNN-based camera model identification used non-adaptive hand-designed linear residuals as classification features. In this part of experiments, we compare the robustness of our ACFM-based CNN to a non-adaptive linear residual extractor, i.e., HPF-based CNN. To accomplish this, we use the NonACFM-based CNN architecture in red dashed line in Fig. 1 where the "Constrained Conv" layer is replaced with the same HPF used in [9].

From Table 1, one can notice that our proposed ACFM-based CNN is significantly more accurate and robust than the non-adaptive HPF-based CNN approach. Additionally, experimental results show that the adaptive NonACFM-based approach is also better than the HPF-based CNN and can achieve an identification rate which is typically higher than 90% accuracy with all possible tampering operations. More specifically, it can achieve 98.52% identification rate with unaltered images and at least 76.05% with 50% downscaled then post-compressed (QF=90) images using an ET classifier. This demonstrates the ability of the constrained convolutional layer to adaptively extract low-level pixel value dependency features directly from data even when input images have undergone a single or multiple tampering operations.

From Table 1, one can notice that the ET classifier has also significantly improved the camera model identification rate of CNN with all underlying processing operations for the HPF-based approach. The HPF-based approach can achieve only 97.83% identification rate with unaltered images and at least 67.42% with 50% downscaled then post-compressed (QF=90) images. The HPF approach achieves a lower identification rate since it is a suboptimal solution of the trained network with a constrained convolutional layer.

5. CONCLUSION

In this paper, we have proposed a new robust deep learning approach to forensically determine the make and model of a camera that captured post-processed image. To accomplish this, low-level pixel-value dependency feature maps learned by a constrained convolutional layer are augmented using the nonlinear MFR features. We evaluated the effectiveness of our proposed approach using the Dresden database, which consists of 26 camera models, on unaltered and altered images created by seven different types of commonly used image processing operations. When subject images are 50% downscaled with and without JPEG post-compression, our proposed ACFM-based CNN significantly outperforms its homologue networks which do not make use of the MFR features.

6. REFERENCES

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade." *IEEE Access*, vol. 1, pp. 167–200, 2013.

[2] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 1. IEEE, 2004, pp. 709–712.

[3] A. Swaminathan, M. Wu, and K. J. R. Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 91–106, 2007.

[4] H. Cao and A. C. Kot, "Accurate detection of demosaicing regularity for digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 4, pp. 899–910, 2009.

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

[6] C. Chen and M. 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.

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

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

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

[10] L. Pibre, P. Jérôme, D. Ienco, and M. Chaumont, "Deep learning for steganalysis is better than a rich model with an ensemble classifier, and is natively robust to the cover source-mismatch," *arXiv preprint arXiv:1511.04855*, 2015.

[11] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2016, pp. 5–10.

[12] X. Kang, M. C. Stamm, A. Peng, and K. R. Liu, "Robust median filtering forensics using an autoregressive model," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 9, pp. 1456–1468, 2013.

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

[14] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.

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

[16] B. Bayar and M. C. Stamm, "A generic approach towards image manipulation parameter estimation using convolutional neural networks," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2017.

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

[18] 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.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

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

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