https://doi.org/10.2352/ISSN.2470-1173.2020.4.MWSF-078

A CNN-Based Correlation Predictor for PRNU-Based Image Manipulation Localization

Sujoy Chakraborty; Stockton university; Galloway, NJ/USA

Abstract

For PRNU-based forensic detectors, the fundamental test statistic is the normalized correlation between the camera fingerprint and the noise residual extracted from the image in successive overlapping analysis windows. The correlation predictor plays a crucial role in the performance of all such detectors. The traditional correlation predictor is based on predefined hand-crafted features representing intensity, texture and saturation characteristics of the image block under inspection. The performance of such an approach depends largely on the training and test data. We propose a convolutional neural network (CNN) architecture to predict the correlation from image patches of suitable size fed as input. Our empirical finding suggests that the CNN generalizes much better than the classical correlation predictor. With the CNN, we could operate with a common network architecture for various digital camera devices as well as a single network that could universally predict correlations for content from all cameras we experimented with, even including the ones that were not used in training the network. Integrating the CNN with our forensic detector gave state-of-the-art results.

Introduction

Digital camera sensor noise has been accepted as one of the most important and valuable characteristics in the forensics research community [1]. A range of seminal works by Fridrich et. al. [2] establishes the fact that a unique spatially varying noise pattern characterizes the sensor of each camera. Due to imperfections in the silicon wafer and inconsistencies in the manufacturing process, the resulting intensity at every pixel varies when the sensor is illuminated homogeneously. This gives rise to the sensor noise pattern which is a unique fingerprint that is present in every picture taken by the camera. This is also called the Photo Response Non-uniformity or the PRNU noise and can be exploited for device identification as well as manipulation localization. For copy-move or splicing manipulations, the PRNU gets distorted by the manipulated content in the image. A straightforward manipulation localization algorithm inspects the query image in small analysis windows and compares the local noise estimate to the corresponding part of the camera fingerprint in terms of the normalized correlation score. The content is marked as manipulated if the test statistic falls below a suitably chosen threshold [7]. Recently, the spatial dependency between neighboring pixels has been exploited to improve the performance of the fundamental detector [3, 4, 5].

The main challenge that the correlation-based sliding window detectors face is that the measured correlation strongly depends on the image content. Dark, textured and saturated regions in an image are likely to yield a low correlation score even in the absence of manipulation. The correlation predictor has been proposed to mitigate this challenge [7]. Based on local image features representing the texture, intensity and saturation characteristics, it predicts the expected value of the correlation score if the content is genuine. The correlation predictor originally proposed by Chen et. al. [7] is a least square fit to the quadratic expansion of features representing the texture, intensity and saturation characteristics. Recently, Korus et. al. [4] trained a feed forward neural network with five hidden layers with the same features proposed in [7].

In this paper, we propose a correlation predictor based on a convolutional neural network (CNN) architecture. We particularly emphasize the idea of following a data-driven approach, allowing a deep neural network to automatically learn features which are useful for predicting the correlation, rather than relying on hand-crafted features measuring the texture, intensity and saturation attributes of the content. We demonstrate the efficacy of the CNN by proposing a network architecture that outperforms the performance of the feed-forward neural network predictor [4]. Along the way, we also demonstrate the power of the CNN to operate on a universal setup, where we predict correlation very accurately for patches from cameras not used for training the CNN.

In the rest of the paper, we review the classical correlation predictor in the next section. We then discuss the literature on common applications of CNN in image forensics. Subsequently, we describe in detail the architecture of our proposed network. We follow the architectural description with our experimental setup and the detailed training procedure of the network. Subsequently, we present manipulation localization results with the classical correlation predictor as well as the proposed CNN predictor, with our manipulation localization algorithm [5], followed by concluding the paper.

Correlation Predictor

The correlation predictor has been defined as a mapping from a feature space to a real number. In the literature of image forensics, typically it has been expressed as a linear combination of terms in a quadratic expansion of features quantifying the intensity, texture and saturation characteristics [7]. It is basically the prediction of the normalized correlation $\rho_b = corr(\mathbf{x_b}, \mathbf{w_b})$ observed on small blocks of the image, assuming the content to be genuine. Here, $\mathbf{x_b}$ and $\mathbf{w_b}$ respectively denote the PRNU signal and the sensor noise measured in the b-the block in the image. Following is a brief description of the features used in the classical correlation predictor [7].

Image Intensity

Due to the reason that the PRNU term is multiplicative, the correlation is higher in areas of high intensity. However, the PRNU term is not present in saturated regions. Hence, for all sites with intensity above a critical value γ, the intensity is attenu-

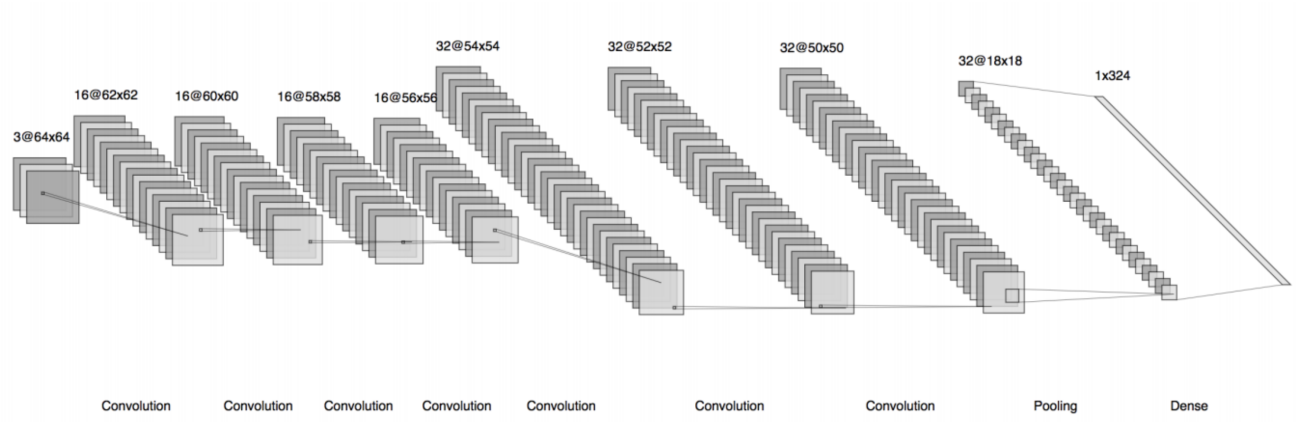


Figure 1: Proposed architecture of the convolutional neural network used to predict correlation.

ated. The intensity feature has been defined as the average image intensity attenuated close to the maximum dynamic range, which corresponds to the range $\gamma \leq x_i < 255$, where x_i denotes the i-th pixel in the image in column-major order. Mathematically:

$$f_I = \frac{1}{|B|} \sum_{i \in B} att(x_i), \tag{1}$$

where, the attenuation function $att(x)$ is defined as:

$$att(x) = \begin{cases} \exp\left(-(x-\gamma)^2/\delta\right) & \text{if } x > \gamma \\ x/\gamma & \text{if } x \leq \gamma \end{cases} \tag{2}$$

In equation (2), $|B|$ denotes the number of pixels in the block B and the parameters γ and δ vary for different camera models.

Texture

The correlation tends to be lower in textured regions. Since the textured regions correspond to high frequency, some traces of the textured content is also present along with the noise residual due to imperfect filtering which weakens the PRNU signal, which is essentially our signal of interest. The texture feature is defined as follows:

$$f_T = \frac{1}{|B|} \sum_{i \in B} \frac{1}{1+\sigma_i{}^2}, \tag{3}$$

Here, σ_i refers to the standard deviation of the image in a 5×5 window around the i-th pixel in the image in column-major order.

Signal Flattening

The predictor would overestimate the correlation in a relatively flat and high intensity unsaturated area, with a low value of local variance. The saturation feature is defined as the fraction of pixels in a block with average local standard deviation below a threshold. Mathematically:

$$f_S = \frac{1}{|B|} |\{i \in B, \sigma_i < \eta x_i\}|, \tag{4}$$

Here, η is a constant which depends on the variance of the PRNU and σ_i is the local standard deviation of the image intensity of a block of reasonable size around the i-th site in the image block.

Texture-Intensity

The measured correlation also strongly depends on the collective influence of texture and intensity. Sometimes, highly textured regions are also high-intensity regions. The Texture-Intensity feature measures the combined effect of texture and intensity in an image and is defined as:

$$f_{TI} = \frac{1}{|B_b|} \sum_{i \in B_b} \frac{att(x_i)}{1+\sigma_i{}^2} \tag{5}$$

where, $att(x)$ and σ_i has meanings as defined before.

The correlation predictor has been modeled as a linear combination of these features and their second order terms [7], where the coefficients of the polynomial are obtained through a least square regression fit. Korus et. al. [4] used the same features to train a feed-forward neural network with five hidden layers. We found the predictor proposed in [4] to be more effective than the least square fit proposed in [7], so we omit the discussion on the least-square fit here. We choose the predictor proposed by Korus as our benchmark to compare our CNN-based predictor, both in terms of the prediction error as well as manipulation localization performance with our detector [5].

Deep Learning in Image Forensics

Deep learning in computer vision problems has seen remarkable success applying the convolutional neural network (CNN) in problems like image classification [19], image captioning [20] etc. Given a sufficiently large training set, CNN-s are capable to learn the best features automatically from the data for the given task. The CNN has already been successfully applied in image forensics [8, 9, 10, 11, 12] as well as in steganalysis of JPEG images[13, 14]. In [8], the authors propose a CNN to automatically learn hierarchical representations from input color images, specifically for the detection of splicing and copy-move manipulations. In [9], the authors introduce a new form of convolutional layer in the proposed CNN architecture to suppress image content and adaptively learn manipulation detection features. In [10], the authors show that a class of residual-based descriptors can be regarded as a constrained CNN. In [11], the authors propose a CNN architecture to classify the type of global processing applied to an image. In [12], the authors propose a CNN-based architecture to extract a camera model fingerprint which they termed as Noiseprint. Thus, the CNN has been applied to a wide variety

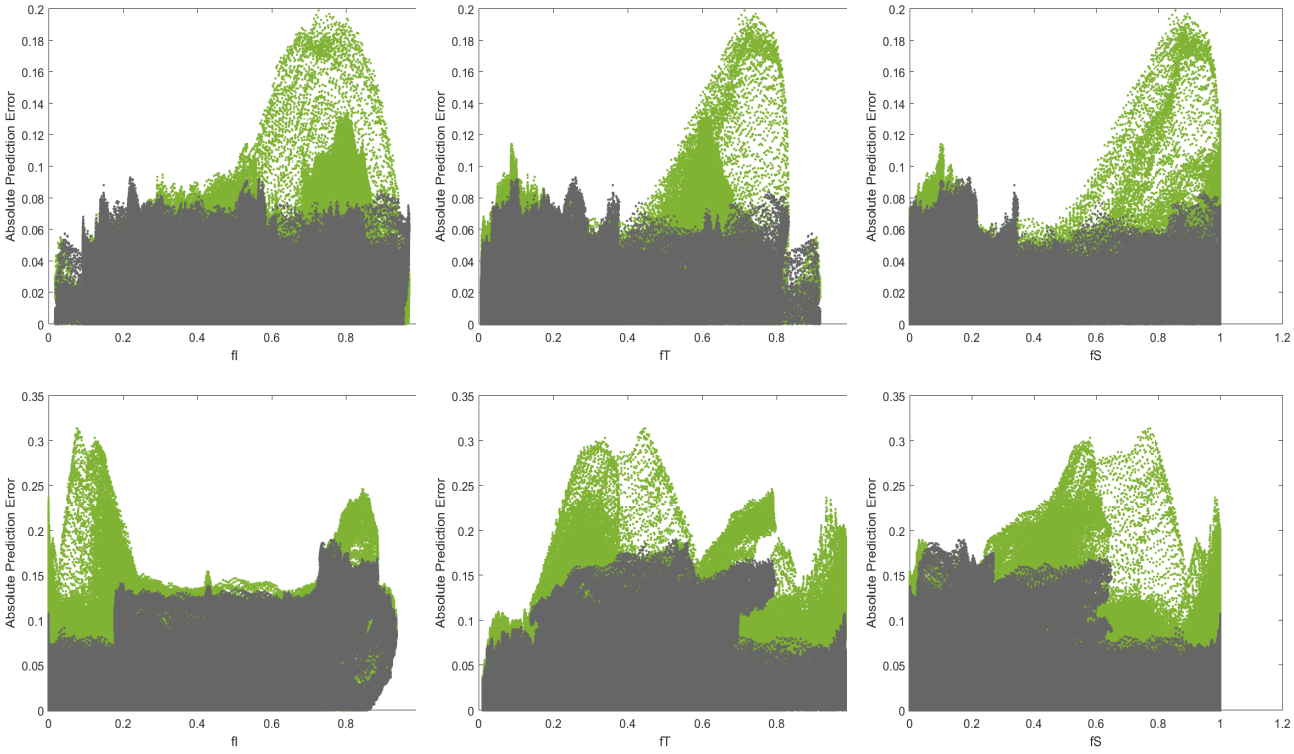


Figure 2: Plot of prediction error vs features characterizing the image content. Left: Intensity feature, Middle: Texture feature, Right: Signal flattening feature. : feature-based, : CNN

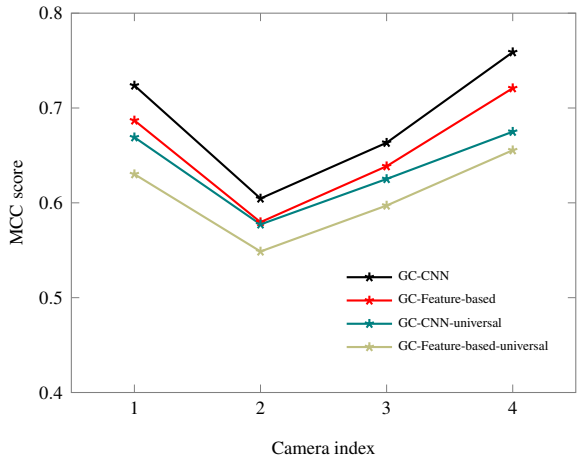


Figure 3: Manipulation localization comparison with different detectors for classical and CNN predictors (Uncompressed images)

of forensic problems which gave us enough motivation to explore if the CNN could work as a correlation predictor. The idea was to let the CNN learn features automatically to predict the correlation depending upon the image content, rather than relying on hand-crafted features. Our goal was to come up with a common network architecture that would work substantially well for all camera models.

Proposed CNN architecture

We propose a convolutional neural network architecture to predict the local correlations at every pixel in an image. Our network architecture consists of 7 un-pooled convolutional layers followed by a pooling layer and a fully connected layer. We follow some of the design choices which were successful in the works [14] and [15]. In [15], the authors proposed a CNN-based image denoiser with convolutional filters of size 3×3, without having any pooling layers in between convolutional layers. Also, in [14], the authors use 16×16 pooling size before the fully connected layer. In our proposed network architecture as shown in Figure 1, we have all convolutional layers un-pooled with 3×3 filters and a 16×16 pooling before the fully connected layer. The network that we propose has a total of 7 CONV layers. Among the 7 convolutional layers, the first 5 layers are having 16 filters each and the next 2 layers are having 32 filters each. All filters are of dimension 3×3. We added batch normalization after each convolutional layer, followed by the leaky RELU non-linearity. The leaky RELU function has a small negative slope which prevents stopping of parameter updates during training due to zero gradients [16]. We used max-pooling after the last convolutional layer. A window of size 16×16 was used for pooling. The output from the pooling layer is a set of 32 blocks of size 18×18 which is fed to the fully connected layer. For our benchmark predictor based on the feedforward neural network, we simply had 5 inputs going to the fully connected layer (the 4 features proposed by Chen et. al. [7] and a bias term). The output of the fully connected layer of our network is one single number, which is the

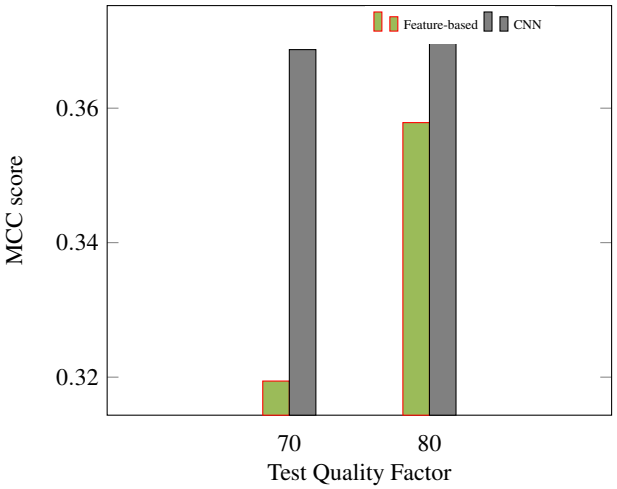


Figure 4: Manipulation localization comparison with different detectors for classical and CNN predictors (JPEG compressed images)

correlation value that the network predicts from the input image block. Batch normalization is used as a preprocessing step after each of the convolutional layer in the network. This process typically normalizes the data to follow a unit Gaussian distribution. It is a necessary step because the distribution of each layer's input changes during training, as the parameter of the previous layer is updated, which slows down the training process [17]. Weights in the convolutional layer are initialized with the random normal initializer. Xavier initializer is used to initialize the weights in the Fully connected layer [18]. In each convolutional layer we use L2 regularization to avoid overtraining of the network, which ensures that the network would generalize better on unseen data. We used patches from randomly chosen untampered images from the Realistic Tampering dataset [4], to train the network for each camera. From each camera, a large set of image blocks were chosen to train the network. To compute the loss function during the training, we also had the known correlation values for each patch used in training the network. We considered patches of dimension 64×64 to train the network. A Mean Square Error (MSE) based loss function was used to compute the loss during the training process. The MSE is defined as the average prediction error for each batch of patches used in training. The prediction error is computed from the known correlation values for each image block of size 64×64.

Experimental Setup

Our experimental results are based on the Realistic Tampering dataset [4] and the Dresden [28] dataset. The Realistic Tampering dataset comprises of 220 uncompressed manipulated images from 4 different camera models, 55 from each camera. There are also 220 corresponding images which are untampered. Also, there are separate images to train the respective camera model (i.e. for fingerprint estimation as well as predictor training). We also created JPEG versions of the Realistic Tampering dataset for quality factors 70, 80 and 90 to perform experiments with JPEG images for various settings. For the Dresden dataset, we worked with 175 uncompressed images from a Nikon D70 camera model. The proposed network was trained to operate on a per-camera setting (i.e. training a separate predictor for each camera), as well as a universal setting (a single predictor for all cameras).

Training the network separately for each camera model

First we trained the network for each camera in the Realistic Tampering dataset [4]. We used patches from the untampered images, where each patch is of size 64×64 pixels. Images were converted to greyscale before sampling the patches. For all the image patches, we also had the corresponding normalized correlation value computed between the patch noise residue and the corresponding patch from the fingerprint. The training set thus consists of known image patches and corresponding correlation values. For training with uncompressed images, we randomly selected 10^5 blocks (about 5 percent of the total 1080×1920 blocks per image) from each of the 55 pristine images in the dataset. We also trained the network separately with JPEG images compressed with quality factors of 70, 80 and 90, to test against the respective quality factors. The JPEG images used for training were images which were meant to be used explicitly for training the camera models in the Realistic Tampering dataset. After the training of the network, we performed manipulation localization with our DRF detector [5] on both uncompressed images as well as JPEG compressed images of the above mentioned quality factors.

Training a single network for all camera models

We were also able to achieve state-of-the-art manipulation localization performance with a single CNN predictor which is trained on patches from all cameras of the Realistic Tampering dataset. This is a promising result considering the fact that separate training is needed for every digital camera when we work with the feature-based predictor proposed by Chen et al. [7], to achieve an equivalent manipulation localization performance for any PRNU-based detector. To train a common predictor with the CNN, we used a greyscale patch from the image in first channel, the corresponding patch from the noise residue in the second channel and the corresponding patch from camera fingerprint in the third channel. Hence, the input to the network has three channels in this case and is of dimension $64 \times 64 \times 3$. We used $25K$ patches from each camera model of the Realistic Tampering dataset, for a total of 10^5 patches for training the network. It should be noted here that we have 3 channels in this case, which are not RGB channels, but we feed the greyscale image patch in channel - 1, corresponding patch residual in channel - 2 and corresponding patch from PRNU in channel - 3. For the per camera predictor, we only used the greyscale patches for training (hence only one single channel).

We measured the performance of each predictor in terms of the absolute prediction error, as well as the performance of manipulation localization in terms of MCC score, when we integrate each predictor with our image manipulation localization algorithm [5]. The MCC score is defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{6}$$

where, TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

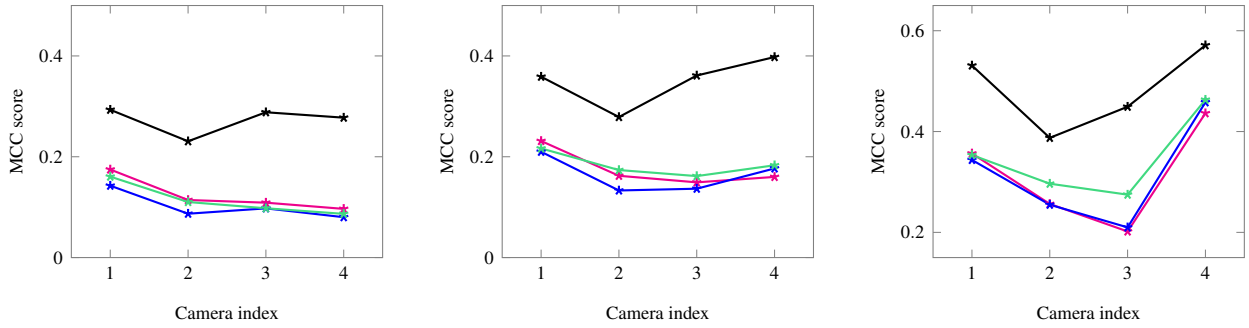


Figure 5: Manipulation localization performance with CNN for JPEG images. Predictors trained with Left: Q = 70, Middle: Q = 80, Right: Q = 90. Algorithms: Adaptive window —*—, Segmentation guided —*—, Multiscale fusion —*—, DRF Graphcut (CNN) —*—

Experimental Results

In Figure 2, we demonstrate the efficacy of the CNN with some examples from the Realistic Tampering dataset, with the plot of absolute prediction error against the features characterizing the intensity, texture and saturation for the two predictors. The CNN predictor is the one obtained from the universal setup, where patches from all 4 cameras were used from the Realistic Tampering dataset for training the network. For the feature-based predictor, we trained the predictor 10 times, each time with a different set of $30K$ image patches. Among the 10 trained predictors per camera, we picked the one which gave the best overall MCC score of manipulation localization, separately for each camera. The feature-based predictor was trained with known features computed for images patches of dimension 64×64. We observe that the CNN predictor does a better prediction of correlation than the feature-based predictor and gives an overall lower prediction error. This demonstrates the efficacy of the CNN to generalize over all cameras, which does a better prediction than the feature-based predictor, even under the universal setup. Figure 3 shows the comparison of the two predictors under per camera and universal setup when they are integrated with our manipulation localization algorithm [5]. The results shown in Figure 3 are for uncompressed manipulated images from the Realistic Tampering dataset. The comparison is based on MCC score for manipulation localization results. For all the predictors, the same detector have been used [5]. We observe that the CNN predictor clearly outperforms the feature-based predictor when we have a predictor for each individual camera we work with, as well as when we have a universal predictor trained with patches from all camera devices in the dataset. We found that the CNN does considerably better than the feature-based predictor, which demonstrates the efficacy of the CNN to operate under a universal setup and its ability to generalize on unseen data. As expected, we observe a performance drop under the universal setup for both the predictors when compared to the per camera setup. However, it is also interesting to observe that even under the universal setup, the performance of the CNN is very comparable with the best operating point of the feature-based correlation predictor under the per camera setup.

Figure 4 shows the comparison of the two predictors on JPEG images. We found that the CNN is much more effective than the feature-based predictor, especially with test images of lower quality factors. Here we show the result for the Realistic Tampering dataset on test images of quality factors 70 and 80. Both the predictors were trained separately on a quality factor of 70 in this case. For test images with higher quality factors, even with a small training set, we found the CNN to give comparable results as the feature-based predictor.

Figure 5 shows some more results with JPEG compressed manipulated images from the Realistic Tampering dataset. Here, we compare the performance of our manipulation localization algorithm with three other state-of-the-art manipulation localization algorithms proposed by Korus.et.al. [4]. The predictor used for our detector [5] is the CNN and the predictor used for algorithms proposed by Korus is the feature-based predictor. For each quality factor Q, the predictor as well as the manipulated images are of the same quality factor Q. For this experiment, we considered $Q = 70$, $Q = 80$ and $Q = 90$ respectively. We observe that our algorithm with the CNN predictor clearly outperformed all the detectors proposed in [4], for all camera models in the Realistic Tampering dataset, for all the three quality factors. Figure 6 shows the predicted correlation maps obtained with the CNN as well as the feature-based correlation predictor, along with the actual correlation map for some manipulated images of Realistic Tampering dataset.

To verify the efficacy of the universal CNN on images from unseen cameras, we used the CNN predictor trained on the Realistic Tampering dataset to test manipulation localization performance on the Dresden dataset. For this purpose, we chose 175 uncompressed images from a Nikon D70 camera in the Dresden dataset and replaced a square region of size 384×384 located at a random position in the image by the content from a different camera. We then crop all images to the size of 1080×1920 pixels. The CNN trained on the uncompressed images from Realistic Tampering dataset under the universal setup was used to predict the correlation for each test image from the Dresden dataset. Figure 7 shows some manipulation localization results from the Dresden dataset with our DRF algorithm [5]. The average MCC score for the tested images was 0.84272.

Conclusion

In this paper, we propose a CNN-based correlation predictor to predict the correlation between the noise residue and the camera fingerprint, which is the fundamental test statistic for all PRNU-based forensic detectors. We operated with a common network architecture for all the camera devices we worked with and trained the network per camera device, as well as a sin-

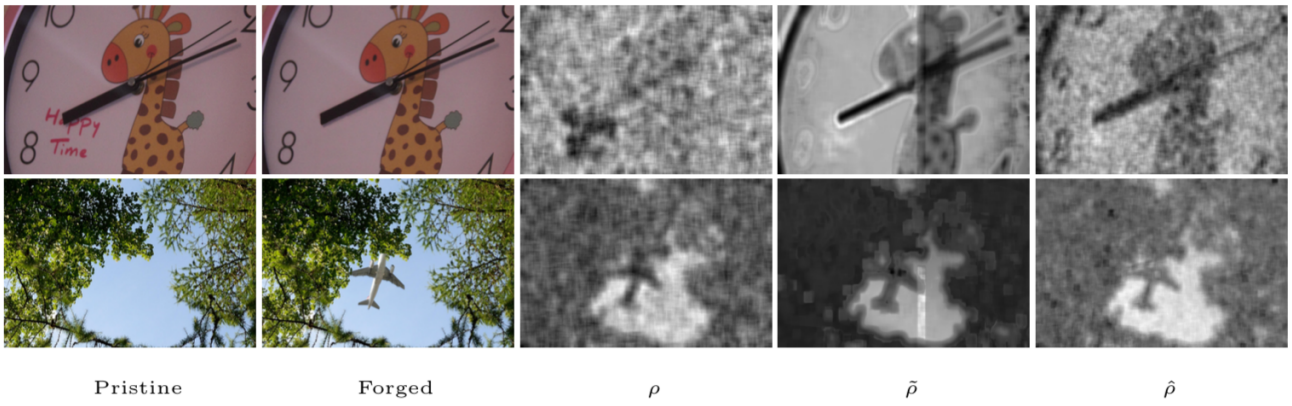


Figure 6: Correlation and predicted correlation maps with different predictors. From left to right: 1. Original image 2. manipulated image 3. actual correlation map 4. predicted correlation map (feature-based) 5. predicted correlation map (CNN). Based on: Realistic Tampering dataset

gle universal predictor for all devices in the dataset. Our motivation was to follow a data-driven approach and automatically learn the features for predicting correlation, rather than relying on hand-crafted features. Our experimental findings suggest that the CNN generalizes better than the feature-based correlation predictor. We obtained better manipulation localization performance with the CNN with our DRF-based manipulation localization algorithm, outperforming the previous best results with the feature-based predictor. We also observed that the CNN does considerably better than the feature-based predictor, especially with JPEG images of lower quality factors. We could also achieve state-of-the-art manipulation localization performance under the universal setup, which gave promising results even on unseen cameras. Our results indicate that the proposed network architecture can be considered as the first stepping stone towards replacing the traditional feature-based correlation predictor with a deep-learning approach.

Acknowledgment

The author would like to thank Pawel Korus for providing reference implementation of his manipulation localization algorithms as well as sharing the images to train camera models in the Realistic Tampering dataset.

References

[1] R. Böhme and M. Kirchner, "Media forensics," in *Information Hiding*, S. Katzenbeisser and F. Petitcolas, Eds. Artech House, 2016, ch. 9, pp. 231–259.

[2] J. Fridrich, "Sensor defects in digital image forensics," in *Digital Image Forensics: There is More to a Picture Than Meets the Eye*, H. T. Sencar and N. Memon, Eds. Springer, 2013, pp. 179–218.

[3] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A Bayesian-MRF approach for PRNU-based image forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 554–567, 2014.

[4] P. Korus and J. Huang, "Multi-scale analysis strategies in PRNU-based tampering localization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 809–824, 2016.

[5] S. Chakraborty, and M. Kirchner, "PRNU-based Image Manipulation Localization with Discriminative Random Fields," in *IS & T Electronic Imaging - Media Watermarking, Security and Forensics*, 2017.

[6] T. Gloe, S. Pfennig, and M. Kirchner, "Unexpected artefacts in PRNU-based camera identification: A 'Dresden Image Database' case-study," in *ACM Multimedia and Security Workshop (MM&Sec)*, 2012, pp. 109–114.

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

[8] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images", in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.

[9] B. Bayar and M. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer", in *ACM Workshop on Information Hiding and Multimedia Security*, 2016.

[10] D. Cozzolino, G. Poggi and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection", in *ACM Workshop on Information Hiding and Multimedia Security*, 2017.

[11] M. Boroumand and J. Fridrich, "Deep learning for detecting processing history of images", in *IS & T Electronic Imaging - Media Watermarking, Security and Forensics*, 2018.

[12] D. Cozzolino and L. Verdoliva, "Noiseprint: a CNN-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2019.

[13] G. Xu, H. Z. Wu and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis", *IEEE Signal Processing Letters*, vol. 23, no. 4, pp. 554–567, 2016.

[14] M. Chen, V. Sedighi, M. Boroumand and J. Fridrich, "JPEG-Phase-Aware Convolutional Neural Network for Steganalysis of JPEG Images", in *ACM Workshop on Information Hiding and Multimedia Security*, 2017.

[15] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising", *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[16] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[17] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep

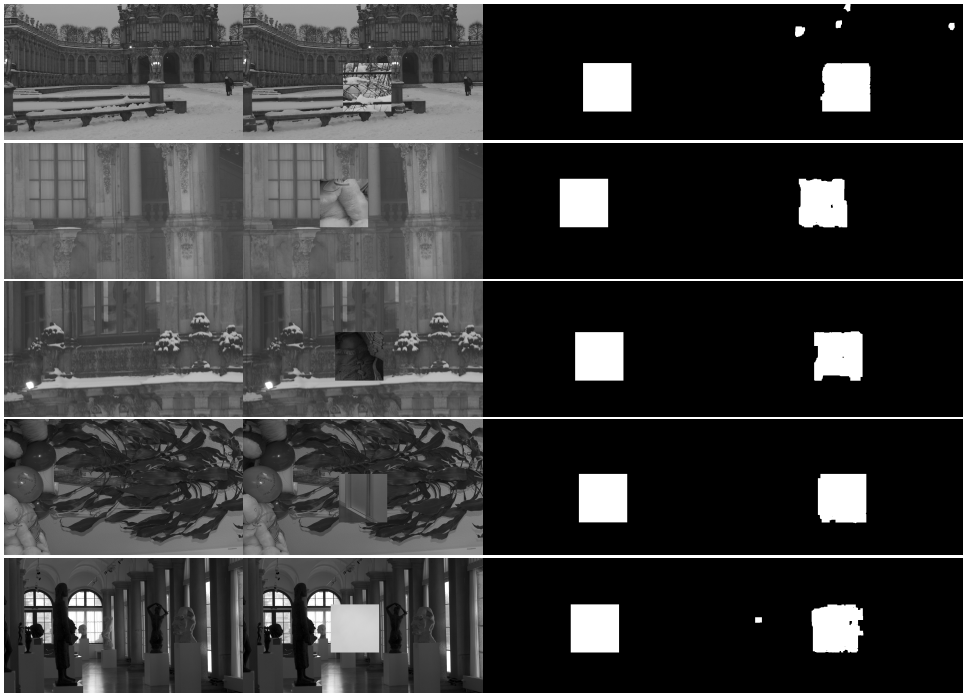


Figure 7: Manipulation localization results on images from Dresden dataset when the CNN is trained on Korus dataset. From left to right: Original image; manipulated image; ground truth; mask generated with the CNN predictor.

Network Training by Reducing Internal Covariate Shift", in *International Conference on Machine Learning (ICML)*, 2015.

[18] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks", in *NIPS*, 2012.

[20] J. Aneja, A. Deshpande, and A. Schwing, "Convolutional image captioning", in *CVPR*, 2018.

[21] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks", in *ECCV*, 2014.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *ICLR*, 2015.

[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in *CVPR*, 2014.

[24] R. Girshick, "Fast R-CNN", in *ICCV*, 2015.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", in *CVPR*, 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *CVPR*, 2016.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", in *NIPS*, 2015.

[28] 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.

[29] M. K. Mıhçak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of Wavelet coefficients," *IEEE Signal Processing Letters*, vol. 6, no. 12, pp. 300–303, 1999.

[30] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6272–6276.

[31] W. Fan, K. Wang, and F. Cayre, "General-purpose image forensics using patch likelihood under image statistical models," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.

Author Biography

Sujoy Chakraborty received his ME in Software Engineering from Jadavpur University, Kolkata (2008), India and his PhD in Electrical and Computer Engineering from the State University of New York, Binghamton (2019). He is currently working as an Assistant Professor of Computer Science at Stockton University, NJ. His work has focused primarily on digital image forensic techniques based on digital camera sensor noise, as well as algorithms based on compression forensics and counter forensics.

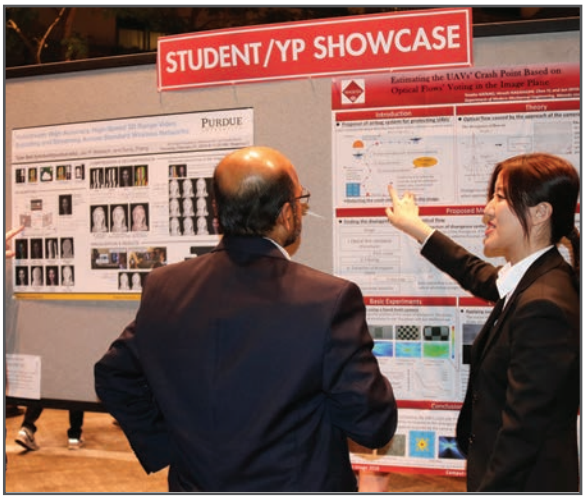
JOIN US AT THE NEXT EI!

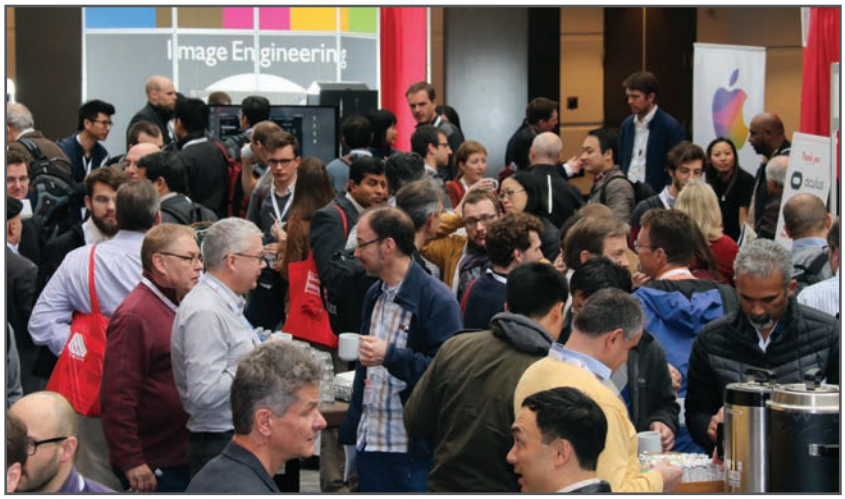
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org