Forensic Scanner Identification Using Machine Learning

Ruiting Shao and Edward J. Delp
Video and Image Processing Laboratory (VIPER)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

***Abstract*—Due to the increasing availability and functionality of image editing tools, many forensic techniques such as digital image authentication, source identification and tamper detection are important for forensic image analysis. In this paper, we describe a machine learning based system to address the forensic analysis of scanner devices. The proposed system uses deep-learning to automatically learn the intrinsic features from various scanned images. Our experimental results show that high accuracy can be achieved for source scanner identification. The proposed system can also generate a reliability map that indicates the manipulated regions in an scanned image.**

***Keywords*-scanner classification; machine learning; media forensics; convolutional neural network;**

I. Introduction

With powerful image editing tools such as Photoshop and GIMP being easily accessible, image manipulation has become very easy. Hence, developing forensic tools to determine the origin or verify the authenticity of a digital image is important. These tools provide an indication as to whether an image is modified and the region where the modification has occurred. A number of methods have been developed for digital image forensics. For example, forensic tools have been developed to detect copy-move attacks [1], [2] and splicing attacks [3]. Methods are also able to identify the manipulated region regardless of the manipulation types [4], [5]. Other tools are able to identify the digital image capture device used to acquire the image [6], [7], [8], which can be a first step in many types of image forensics analysis. The capture of "real" digital images (not computer-generated images) can be roughly divided into two categories: digital cameras and scanners.

In this paper, we are interested in forensics analysis of images captured by scanners. Unlike camera images, scanned images usually contain additional features produced in the pre-scanning stage, such as noise patterns or artifacts generated by the devices producing the "hard-copy" image or document. These scanner-independent features increase the difficulty in scanner model identification. Many scanners also use 1D "line" sensors, which are different than the 2D "area" sensors used in cameras. Previous work in scanner classification and scanned image forensics mainly focus on handcrafted feature extraction [9], [10], [11]. They extract features unrelated to image content, such as sensor pattern noise [9], dust and scratches [10]. In [12], Gou *et al.* extract statistical features from images and use principle component analysis (PCA) and support vector machine (SVM) to do scanner model identification. The goal is to classify an image based on scanner model rather than the exact instance of the image. In [9], linear discriminant analysis (LDA) and SVM are used with the features which describe the noise pattern of a scanned image to identify the scanner model. This method achieves high classification accuracy and is robust under various post-processing (*e.g.* , contrast stretching and sharpening). In [10], Dirik *et al.* propose to use the impurities (*i.e.* , dirt) on the scanner pane to identify the scanning device.

Convolutional neural networks (CNNs) such as VGG [13], ResNet [14], GoogleNet [15], and Xception [16] have produced state-of-art results in object classification on ImageNet [17]. CNNs have large learning capacities to "describe" imaging sensor characteristics by capturing low/median/high-level features of images [8]. For this reason, they have been used for camera model identification [8], [18] and have achieved state-of-art results.

In this paper, we propose a CNN-based system for scanner model identification. We will investigate the reduction of the network depth and number of parameters to account for small image patches (*i.e.* , 64 × 64 pixels) while keeping the time for training in a reasonable range. Inspired by [16], we propose a network that is light-weight and also combines the advantages of ResNet [14] and GoogleNet [15]. The proposed system can achieve a good classification accuracy and generate a reliability map (*i.e.* , a heat map, to indicate the suspected manipulated region).

II. Proposed System

The proposed system is shown in Figure 1. An input image I is first split into smaller sub-images I_s of size $n \times m$ pixels. This is done for four reasons: a) to deal with large scanned images at native resolution, b) to take location independence into account, c) to enlarge the dataset, and d) to provide low pre-processing time and memory usage.

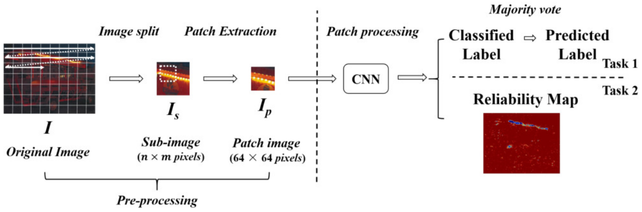
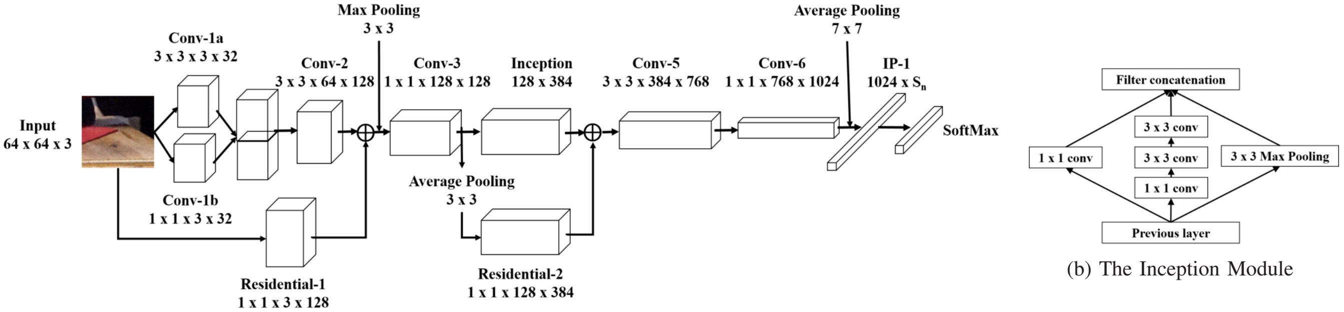


Figure 1: Our approach for scanner model classification.

A. Training

As indicated in Figure 1, input image I is split into sub-images I_s ($n \times m$ pixels) in zig-zag form. The values of n and m should be no smaller than 64. From each I_s, a patch of size 64 × 64 is extracted from a random location. We denote this extracted patch as I_p. These extracted patches I_p along with their corresponding labels S are inputs into the network. This pre-processing enables the proposed system to work with small-size images and use a

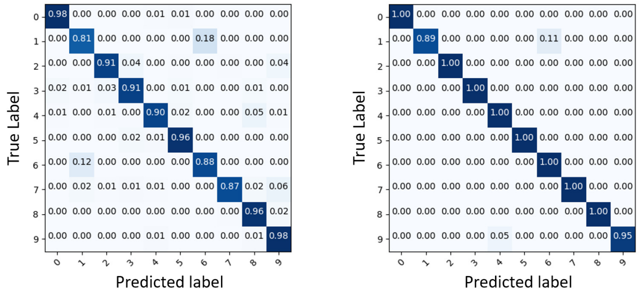


(b) The Inception Module

(a) Our Proposed Network

Figure 2: (a) is the neural network architecture investigated in this paper. The input image size is 64×64 pixels. The "IP layer" is the inner product layer. The residential block is a 1×1 convolution layer followed by batch normalization. The SoftMax layer acts as a normalized exponential function applied to the layer before the output. The output of SoftMax layer can be interpreted as the probabilities of the input image patch belonging to the scanner models individually. The final output layer size is $1 \times N_s$, where N_s is the number of the scanner models used in the training stage. (b) is the inception module used in (a).

smaller network architecture to save training time and memory usage. Designing a suitable network architecture is an important part in the scanner model identification system. There are several factors that need to be considered to build the network: a) the kernel size, b) the utilization of pooling layers, c) the depth of the network, and d) the implementation of the network modules. Our proposed network is shown in Figure 2.



True Label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	0.98	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
1	0.00	0.81	0.00	0.00	0.00	0.00	0.18	0.00	0.00	0.00
2	0.00	0.00	0.91	0.04	0.00	0.00	0.00	0.00	0.00	0.04
3	0.02	0.01	0.03	0.91	0.00	0.01	0.00	0.00	0.01	0.00
4	0.01	0.00	0.01	0.00	0.90	0.02	0.00	0.00	0.05	0.01
5	0.00	0.00	0.00	0.02	0.01	0.96	0.00	0.00	0.00	0.00
6	0.00	0.12	0.00	0.00	0.00	0.00	0.88	0.00	0.00	0.00
7	0.00	0.02	0.01	0.01	0.01	0.00	0.00	0.87	0.02	0.06
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.02
9	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.98

(a) per patch

True Label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.89	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
9	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.95

(b) per image

Figure 3: Confusion matrices for the 10-scanner dataset. Each cell indicates the classification accuracy of predicated labels based on true labels.

B. Testing

The same pre-processing procedure as described in the training section will be used in the testing stage. A test image will first be split into sub-images, and then subsequently extracted into patches of size 64×64 pixels. The extracted patches will be used as inputs for the proposed neural network.

As shown in Figure 1, our proposed system will evaluate two tasks on scanned images: scanner model classification and reliability map generation. In Task 1 (scanner model classification), we assign the predicted scanner labels to both patches I_p and original images I. The predicted scanner label for the sub-image I_s is the same as the predicted label of its corresponding patch I_p. The classification decision for the original image I is obtained by majority voting over the decisions corresponding to its individual sub-images I_s. In Task 2, a reliability map [19] is generated based on the majority vote result from Task 1. The pixel values in the reliability map indicate the probability of the corresponding pixel in the original image being correctly classified. The probability of pixel x belonging to scanner s is the average value of the corresponding probabilities for the sub-images that contain pixel x:

$$\mathcal{P}_s(x) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{P}_s(Sub_i) \tag{1}$$

where Sub_i indicates the sub-image including pixel x, n indicates the total number of these sub-images, and $\mathcal{P}_s(\cdot)$ indicates the probability of $\cdot$ belongs to scanner s.

III. The Experiments

In this section, we describe the dataset we use and the experiments conducted by using the proposed system in Figure 1.

A. Dataset

We use the Dartmouth Scanner Dataset for our experiments.[1] This dataset contains a total of 3,874 scanned images in JPEG format from 169 different scanner models. The size of the original scanned images varies from 500×500 pixels to $5,000 \times 5,000$ pixels with various scan resolutions (dpi - dots per inch). For each scanner model, we randomly partition its images into three subsets with the ratio of 6:1:3 for training, validation, and testing, respectively. We first construct a small sub-dataset with 10 randomly selected scanners, known as the "10-scanner dataset", to evaluate the performance of the proposed system. We then use the entire dataset to check whether the system is able to scale to a larger dataset.

We also constructed several forged images using a copy-move attack for evaluating our reliability maps. The copied areas are from the same image ("self copy") or from a different image in the dataset ("multi-source copy').

[1] We like to thank Professor Hany Farid for making this dataset available.

Network		10 scanners	169 scanners
Ours	per image	96.83%	92.97%
	per patch	93.72%	89.69%
Xception [16]	per image	95.24%	93.24%
	per patch	92.11%	88.85%
Inception3 [20]	per image	94.44%	90.37%
	per patch	91.69%	88.62%
Resnet34 [14]	per image	96.03%	91.67%
	per patch	91.72%	88.73%

Table I: The scanner model classification accuracy: "per patch" rows indicate the classification accuracy on patches I_p; "per image" rows indicate the classification accuracy for full size images I.

B. *Experimental Results*

Task 1 — Scanner model classification. Our neural network (Figure 2a) is implemented in Pytorch using stochastic gradient descent (SGD) with learning rate 0.01, momentum 0.5 and weight decay 0.0001. We compare our method with some other CNN architectures, such as InceptionV3 [20], Resnet34 [14] and Xception [16].

Figure 3 reports our results in terms of the confusion matrices for the 10-scanner dataset. The overall classification accuracy is 93.72% per patch (i.e. without majority vote) and 96.83% per image (i.e. with majority vote). The high accuracies on patch-level and image-level classification tasks indicate our model is very effective on the 10-scanner dataset. The results for both the 10-scanner dataset and the entire dataset are reported in Table I. On the 10-scanner dataset, our method achieves the highest classification accuracy on both patches and the images. On the entire dataset, our method achieves the highest patch-level accuracy. Our image-level classification accuracy is very close to the highest, the one which achieved by Xception. It must be noted that our model has fewer parameters and is shallower compared to the other CNN architectures.

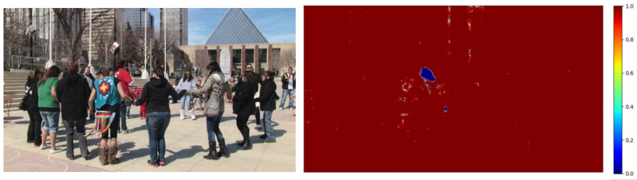


Figure 4: An original scanned image used for forged image creation and its corresponding reliability map with sub-images size set to be 64 × 64 pixels and stride set to be 4 pixels.

Task 2 — Reliability Maps. Since our system is aimed at extracting intrinsic features of scanner models, it should also be able to identify manipulated region irrespective of image content. In this task, we investigate to generate a reliability map (*i.e.* a heat map) that can indicate suspicious forged areas in the images. The reliability map is generated based on the predicted label obtained by majority vote, as explained in equation 1.

Figure 4 shows an example of the reliability map. In the reliability map, the color of the pixel represents the probability that it is generated by the predicted scanner model. Color "dark red" indicates a probability value equal to 1.0, and color "dark blue" indicates a probability value equal to 0.0. Then we use the original image from Figure 4 to generate manipulated images in Photoshop. The forged images are shown in the first column in Figure 5. The top one is generated by self-image copy-move with translation operations. The bottom one is generated by copy-pasting regions in an other image source from different scanner model. The reliability maps generated with different stride size for these two forged images are shown in Figure 5. In the reliability map, the color of the forged region in reliability map is generally bluish. Especially when the forged region is copied from another image source. For the untamped area, some pixels' value in reliability map is between 0.4–0.6 (*i.e.* , shown in light green or yellow). This may be due to the intrinsic features of scanner model for these locations have been changed due to the encoding and decoding in image editing tools. These results indicate the effectiveness of using our reliability map to indicate the suspicious forgery.

IV. Conclusion

In this paper we investigate the use of deep-learning methods to address scanner model classification and localization. Compared with classical methods, our proposed system can: a) learn intrinsic scanner features automatically; b) have no restrictions on data collection; c) associate small image patches (64 × 64 pixels) to scanner models with high accuracy; and d) detect image forgery and localization on small image size. Our experimental results shown in Table I indicate that the proposed system can automatically learn the inherit features to differentiate scanner models and is robust to JPEG compression. The results in Figure 5 show the ability of the proposed system to identify suspected forged regions in scanned images. These experimental results indicate that our reliability map provides a way to detect forgeries in scanned images.

Further work will be devoted to: a) improve the neural network architecture in the proposed system, b) detect other types of forgeries using the proposed system and c) evaluate the performance of the proposed system on scanned documents.

V. Acknowledgments

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. 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 DARPA, AFRL or the U.S. Government. Address all comments to Edward J. Delp, ace@ecn.purdue.edu.

References

[1] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images," *Proceedings of the Digital Forensic Research Workshop*, August 2003, Cleveland, OH.

[2] Sevinc Bayram, Husrev Taha Sencar, and Nasir Memon, "An efficient and robust method for detecting copy-move forgery," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1053–1056, April 2009, Taipei, Taiwan.

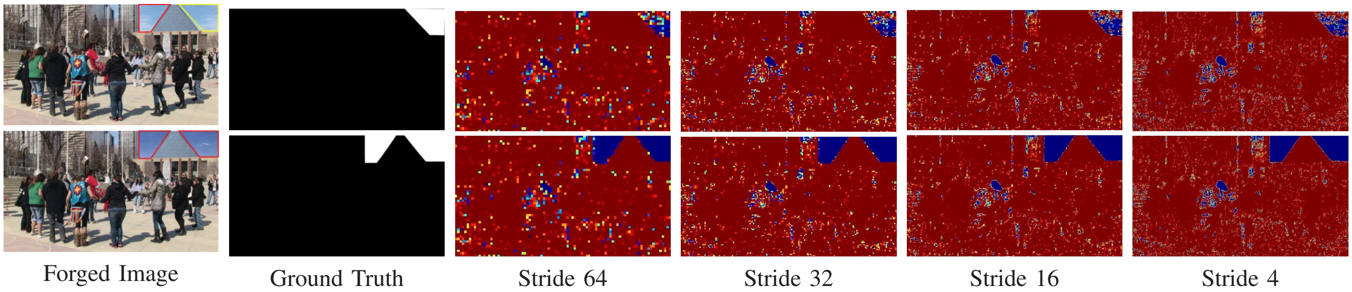


Figure 5: The forged scanned images and corresponding reliability maps with different strides. (i) For the self copy forged image (top), the yellow box region is duplicated from the red box region with horizontal flipping, stretching and compressing operation; (ii) For the multi-source forged image (bottom), the red box region is duplicated from another image scanned by different scanner. In the reliability map, the blue pixels indicate that this region has a high probability of being manipulated. As the reliability maps shows, the smaller the stride, the reliability map can achieve better localization.

[3] Y. Q. Shi, C. Chen, and W. Chen, "A natural image model approach to splicing detection," *Proceedings of the 9th workshop on Multimedia & Security*, pp. 51–62, September 2007, Dallas, TX.

[4] A. C. Popescuand H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, October 2005.

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

[6] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[7] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," *Proceedings of the IEEE International Conference on Image Processing*, pp. 69–72, September 2005, Genova, Italy.

[8] A. Tuama, F. Comb, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, December 2016, Abu Dhabi, United Arab Emirates.

[9] N. Khanna, A. K. Mikkilineni, and E. J. Delp, "Scanner identification using feature-based processing and analysis," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 123–139, March 2009.

[10] A. E. Dirik, H. T. Sencar, and N. Memon, "Flatbed scanner identification based on dust and scratches over scanner platen," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1385–1388, April 2009, Taipei, Taiwan.

[11] T. Gloe, E. Franz, and A. Winkler, "Forensics for flatbed scanners," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, p. 65051I, February 2007, San Jose, CA.

[12] H. Gou, A. Swaminathan, and M. Wu, "Robust scanner identification based on noise features scholar," *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, p. 65050S, February 2007, San Jose, CA.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proceedings of the International Conference on Learning Representations*, May 2015, San Diego, CA.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016, Las Vegas, NV.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, June 2015, Boston, MA.

[16] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800–1807, July 2017, Honolulu, HI.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, JUne 2009, Miami Beach, FL.

[18] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 259–263, March 2017.

[19] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, June, Las Vegas, NV.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, June 2016, Las Vegas, NV.