Estimation of Copy-sensitive Codes Using a Neural Approach

Rohit Yadav, Iuliia Tkachenko, Alain Trémeau, Thierry Fournel

Estimation of Copy-sensitive Codes Using a Neural Approach

Rohit Yadav, Iuliia Tkachenko, Alain Trémeau, Thierry Fournel
Laboratoire Hubert Curien, UMR CNRS 5516, Université de Lyon, UJM-Saint-Etienne
Saint-Etienne, France
fournel@univ-st-etienne.fr

ABSTRACT

Copy sensitive graphical codes are used as anti-counterfeiting solution in packaging and document protection. Their security is funded on a design hard-to-predict after print and scan. In practice there exist different designs. Here random codes printed at the printer resolution are considered. We suggest an estimation of such codes by using neural networks, an in-trend approach which has however not been studied yet in the present context. In this paper, we test a state-of-the-art architecture efficient in the binarization of handwritten characters. The results show that such an approach can be successfully used by an attacker to provide a valid counterfeited code so fool an authentication system.

CCS CONCEPTS

• **Security and privacy** → *Authentication*; • **Computing methodologies** → Neural networks.

KEYWORDS

copy-sensitive codes; authentication; estimation attack; neural networks for binarization; print-and-scan process

ACM Reference Format:
Rohit Yadav, Iuliia Tkachenko, Alain Trémeau, Thierry Fournel. 2019. Estimation of Copy-sensitive Codes Using a Neural Approach. In *ACM Information Hiding and Multimedia Security Workshop (IHMMSec '19), July 3–5, 2019, TROYES, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3335203.3335718

1 INTRODUCTION

Counterfeits is one of the main problems of these days. The number of counterfeited products (luxury goods, medicines, tickets, administrative documents) is predicted to increase three percent each year according to the association for packaging and processing technology. A big amount of security techniques exists such as hashing techniques, watermarking techniques, holograms and security printing techniques. However, most of these solutions are expensive in production or cannot be efficiently verified by non-specialists without specific devices.

The development of security elements that are based on the use of Measurable But Not Duplicable (MBND) characteristics [9] that are formed during a production process, is a promising path for hardcopy document/packaging authentication. MBND can be physical characteristics (see for instance [18, 27]), or some features of the printing process based either on the printer signature profile [1, 16, 19] or on the use of Copy-Sensitive Graphical Codes (CSGC) [14, 15, 24]. These security elements which are cheap and easy to integrate, offers a user-friendly verification, compatible with standard devices for both production and verification. That is why they correspond to some commercial solutions for packaging protection. Nevertheless, the security aspects of these elements are not fully studied. The security of CSGC is based on the stochastic nature of Print-and-Scan (P&S) process. It can be easily shown that every time when an image is printed and scanned some information is lost in comparison with its digital version. This effect is called the information loss principle [14]. Additionally, each printer and scanner has its own signature [21]. Using these two properties, the authentication test can distinguish the original CSGC (printed once) from copied/counterfeited (printed several times).

In this paper, we focus on the unclonability of CSGC and especially on the estimation attack of such codes using a neural approach. We use some random binary codes for our experiments. We show that a small amount of samples can be efficiently exploited in order to estimate the original structure of the digital CSGC.

The rest of the paper is organized as follows. The authentication system is introduced in Section 2. We discuss possible attacks in Section 3. The decoding process using a neural network is depicted in Section 4. The description of constructed database and the experiments are presented in Section 5. Finally, we conclude in Section 6.

2 AUTHENTICATION SYSTEM

The authentication system using CSGC is illustrated in Fig. 1. The authority center creates the valuable document, puts the generated CSGC into this document and prints it using the dedicated printer.

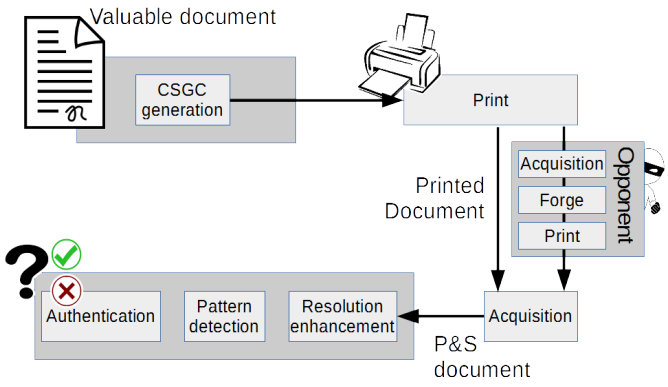
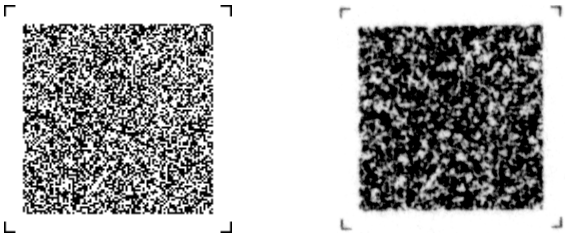


Figure 1: Considered authentication system using CSGC.

IHMMSec '19, July 3–5, 2019, TROYES, France

ACM ISBN 978-1-4503-6821-6/19/06…$15.00
https://doi.org/10.1145/3335203.3335718

During the verification process, the document is scanned and the respective authentication test is applied to verify the authenticity. The generated CSGC is a binary black-and-white image (let call it I), after being printed and scanned this CSGC is a grayscale image ($\tilde{I} = I + N_{PS}$, where N_{PS} is a noise added by P&S process). An example of original (a black-and-white image) CSGC and its degraded (a grayscale image) CSGC versions are illustrated in Fig. 2. The authentication test can be performed as 1) a comparison of the binary CSGC I with an estimated binary code, here denoted as $\hat{I}$ (that is obtained by binarization of grayscale image $\tilde{I}$ during the authentication test) or as 2) a comparison of binary CSGC I with its degraded (graylevel) version $\tilde{I}$. It was shown theoretically [13] that the second strategy is more efficient. The authors in [20] have tested two authentication strategies: i) using Hamming distance between binary original CSGC and binarized versions of printed CSGC, ii) using Pearson correlation between binary original CSGC and graylevel printed CSGC. Nevertheless, the results of ROC curves illustrate that both authentication strategies show the similar results in term of CSGC clonability. Therefore, the comparison of two binary images can give an idea about the attack efficiency.



(a) (b)

Figure 2: Example of a) an original numeric CSGC before printing (I) and b) its degraded version by P&S version ($\tilde{I}$).

In general the authentication test can be formulated as a hypothesis test:

$$\mathcal{H}_0 : d(I; \hat{I}) \leq \epsilon;$$
$$\mathcal{H}_1 : d(I; \hat{I}) > \epsilon,$$

where d is a comparison function (for example, a distance between two images), ϵ is the distortion threshold that was calculated in advance by the authority center.

Most of the attacks are produced between printing and scanning processes. An opponent needs to correctly predict I' - a binarized version of CSGC $\tilde{I}$ that then will be re-printed and re-scanned using the same devices (in the case of the worst type attack) and the authentication test will process a grayscale counterfeited image $\tilde{I}'$. The comparison function d might be sensitive to distortions added during Print-and-Scan (P&S) process and must correctly discriminate CSGC printed ones and CSGC printed twice or counterfeited.

3 POSSIBLE ATTACKS

For counterfeiting, the opponent wants to create copies of CSGC codes as close as possible to original. For this the opponent can use different strategies depending on his/her practical skills. In this section, we discuss about tools that can be used to create counterfeited CSGC codes.

3.1 Uncontrolled duplication

The first most simple way to copy a document is the use of a copy machine or two consecutive P&S operations. However, when we use a copy machine we cannot control the image processing that is applied to the document during the copying process. This attack was explained and experimented in [24, 25]. Instead of using a copy machine, the attack can be performed by the use of a scanner and a printer (the same that were used for the original production in the worst case). In this case, the attack consists of two consecutive P&S operations. This attack can be performed by naive attacker and intuitively will have the same results as the unauthorized duplication using copy machine.

3.2 Smart attack

Here we suppose that the opponent have not only the same reproduction devices but also have several knowledge of image processing and experience in graphics tools. This attack consists also of two consecutive P&S operations, but in this case some basic image processing algorithms are applied between the two P&S operations. The opponent can perform histogram equalization or application of sharpen enhancement of printed and scanned CSGC. These basic image processing techniques can improve the quality of CSGC, but the naive use of image processing techniques [23] cannot help to create good quality counterfeits and thus cannot fool the authentication test.

The authors in [7] propose the so-called "smart" attack: (a) an attacker may try to estimate the original CDP pixels utilizing inverse print-and-scan model; (b) then, s/he can generate genuine-like CDPs and copy CDP protected documents safely. In most experiments the CDP samples resist to such attacks. However, it was shown [2] that an attacker can produce a fake that successfully fools the detector with reasonable number of genuine goods.

This authentication problem can be presented as an optimization game between the legitimate source and an attacker where each player tries to select the best P&S channel to minimize/maximize his authentication performance [11]. For P&S process simulation the lognormal and general Gaussian additive processes [3] are used. The conclusions after studying this minimax game [13]: the opponent optimal parameters are close to the legitimate source parameters for both distribution families and the legitimate source can find a configuration which maximizes its authentication performance.

3.3 Data driven based approach

We can at the end imagine the situation, when an attacker uses devices (printer and scanner) from the same brand and constructs his/her own database of some printed-and-scanned codes as well as of their digital versions. In this case, s/he can estimate the structure of a printed CSGC by using some binarization methods. The use of some classical binarization methods does not give good results and the counterfeited codes cannot pass the authentication test [22]. Authors in [6] suggest to extract some statistics from the printed-and-scanned images then use 5 images in order to train some basic classifiers as LDA, SVM, QDA and naive Bayes. They showed that the use of supervised classification can significantly increase the quality of estimated (counterfeited) codes and the suggested authentication system will be affected by this attack.

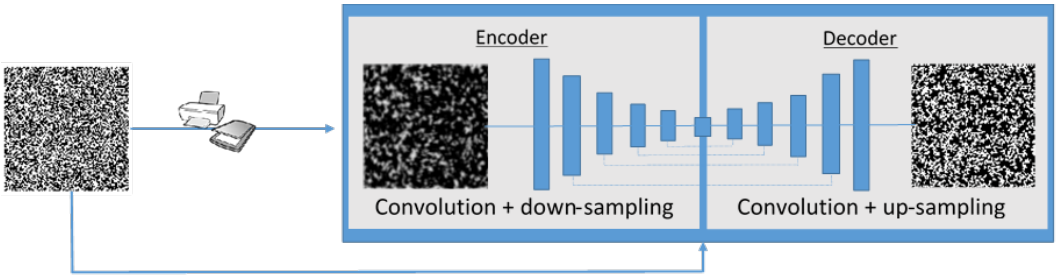


Figure 3: Binarization of CSGC using SAE architecture.

A deep neural network architecture (DNN) has been used in the very recently published paper [20]. In the present paper, we have adopted another type of neural architecture, an autoencoder, in the goal to retrieve a (graphical) code so far decoding CSGC, to study the same authentication system as in [6].

4 EFFICIENT LEARNING FOR DECODING

Auto-encoders are a type of feed-forward neural network which is trained to reconstruct the input as its output. The network has basically two parts: an encoder that learns an encoding function f and a decoder that learns the decoding function g. It has a hidden layer h that provides a useful compressed representation of the input which has been very useful in learning features and for dimensionality reduction [26]. The network basically takes an input x and minimizes the loss function $L(x, g(f(x)))$.

In order to estimate the original code we make use of an existing auto-encoder based architecture called Selectional Auto-Encoder (SAE) [5]. This SAE has previously been used in binarization of document image [5] and has shown significant results. SAE are fully convolutional network without any fully connected layers. It consists of 5 layers of encoder and 5 layers of decoder. The hierarchy of layers in the encoder part of SAE consists of series of convolution and down-sampling operations till the hidden layer h, whereas in the decoder part it consists in series of convolution and up-sampling operations till reconstruction of the image to its original size (see an illustration in Fig. 3).

In a binarization context, SAE is trained to output a selectional value for each pixel from an input image to decide whether that pixel belongs to foreground (black) or background (white). Since SAE performs an image to image processing the classification of each pixel is not computed independently, but also considering the neighbourhood pixels.

The network topology for SAE itself is based on a deep residual Encoder-Decoder Network (RED-Net) [12], which additionally consists of residual connection from its down-sampling layer to its corresponding up-sampling layer. Through these residual connections feature maps learned in the encoding layer are passed to its corresponding decoding layer which helps to recover better clean image through up-sampling. Residual connections have proven to be very effective and lead to better results while training deeper networks [10]. Here, down-sampling is performed by convolution, and up-sampling is achieved using de-convolution. In addition, batch normalization followed by *Relu* activation is added after each convolution and de-convolution layers. The input and output windows of the network were set to 256×256 and size of kernel for convolution and de-convolution was set to 5×5. For all layers 64 feature maps were used. The weights of network were initialized using Xavier uniform initializer [8] while training the network from scratch, otherwise the weights were initialized after pre-training on *DIBCO* database [17]. The learning procedure was carried out using ways of stochastic gradient descent [4] with learning rate of 0.001. The training of the model was done on a GeForce GTX 1080 card for 200 epochs, setting the batch size to 10. However, an early stopping strategy is applied if their is no decrements in the loss after 20 epochs.

Training a SAE consists of providing the input images and its corresponding binary ground-truth of the same size. According to the input size of the network, each image is cropped into several non-overlapping chunks of size 256×256. Once SAE has been completely trained an image can be parsed through the network to obtain its binary version. The output chunks are then recombined to form the full image. This output is of the same size as the input image with restricted value in the range [0,1]. This is due to the use of sigmoid activation in the last layer of the network. After this step a global threshold is still required to obtain the binary values. The pixels whose selectional values are more than the global threshold are labeled as white otherwise as black. It has been shown that the choice of this global threshold does not impact the performance[5]. We have used this implementation from [5] which is available in GitHub [1].

5 EXPERIMENTS

In this section we describe our experiments and discuss the results that we obtained for some decoding methods presented in [6] and the decoding method using the SAE presented in Section 4.

5.1 Database description

In our database[2], we have 100 random binary images with size of 100×100 dots with $48 - 52\%$ of black dots. The real size of printed-and-scanned codes is 4×4 mm. We used a laser printer Xerox Phaser 6500 that has a true 600 dpi (dots per inch) resolution. The printing process was done dot per dot. For acquisition we used a high resolution scanner Epson Perfection V850 Pro with its highest

[1] https://github.com/ajgallego/document-image-binarization

[2] The constructed database is publicly available: www.univ-st-etienne.fr/graphical-code-estimation.

resolution 9600 spi (samples per inch).
The binary images were set in 600 ppi (pixels per inch), the grayscale images had size of 1600×1600 pixels after being printed and scanned with the defined resolutions. So one dot of the image is represented by a square of 16×16 pixels.
We assume in this paper that the authority/authentication center and an opponent have the same printer and scanner. It is the worst case attack, as only the decoding can influence to authentication, but the physical process is the same.
We divide our database into train and test bases. The train and validation database consists of 60 images, the test database consists of 40 images. Depending on the method, the number of samples used for training step changes, nevertheless, the number of test images will be kept the same all the time. We have varied the size of the training database from 60 images to 1 image. We used data augmentation by rotating the images to 90/180/270 degrees and by dividing each of them into four parts. That is why each image is represented by 13 images in the data augmentation setup. The sizes of train databases are presented in Table 2.

5.2 Estimation results

First of all, we implement and test the decoding methods proposed in [6]. The authors mentioned that the best results were obtained with features based on moments (F3 feature set) and with LDA (Linear Discriminant Analysis) classifier. That is why we tested our database only with LDA classifier, but we used all proposed feature vectors (F1-F5). The train base contains of 5 images (as in the reference paper), and we tested these decoding methods on 40 images of our test database. The results are illustrated in Table 1.

Decoding method		BER	Std	Best case	Worst case
LDA	F1	17.14%	0.92%	15.57%	18.86%
	F2	16.72%	0.97%	14.75%	18.64%
	F3	**14.70%**	**0.82%**	**12.97%**	**16.60%**
	F4	16.07%	0.99%	14.31%	17.78%
	F5	16.72%	0.97%	14.75%	18.64%

Table 1: Mean Bit Error Rate comparison for feature sets from [6]. The training sets consist of 5 images.

We compare the 100×100 decoded CSGC with 100×100 digital codes and calculate the Bit Error Rate (BER) for each pair. According to our results the feature set F3 gives the best decoding results with mean BER equal to 14.70%.
Then, we have tested the proposed decoding method based on auto-encoders. To down-sample the parsed 1600×1600 binary output images obtained from our auto-encoders, we apply the majority vote strategy at each 16×16 module of the output binary image in order to consider it as black or a white pixel. The neural network methods are known to be efficient while the database is huge. Nevertheless, in this paper, we would like to find the minimal number of samples that can produce good estimation results. That is why we have tested our approach with $1 - 60$ images. In the same time, as our images are in high resolution, we obtain a quite big number of patches (see Table 2), that is enough to train our SAE.
The results obtained using different size of trained database are presented in Table 3. We note that the better binarization results

Nb of images used	Train database	Number of patches	
		Train	Validation
60 im w/augm.	780 images	14410	50
60 im	60 images	2890	50
5 im w/augm.	65 images	1205	98
5 im	5 images	245	98
1 im w/augm.	13 images	241	98

Table 2: Sizes of trained data sets.

are obtained when we have bigger training database. The BER is 8.75%, when we use 60 images with data augmentation for training. Nevertheless, the binarization results obtained using 1 image with data augmentation are close to the results obtained using 5 images with data augmentation. The BER is 13.38%, when we use 1 image with data augmentation for training, and it is 12.41%, when we use 5 images with data augmentation for training. That means that the high decoding results can be obtained even with one CSGC image in training data base.

Nb of images used	BER	Std	Best case	Worst case
60 im w/augm.	**8.75%**	**0.67%**	**7.49%**	**10.96%**
60 im	9.86%	0.87%	8.32%	12.25%
5 im w/augm.	12.41%	0.99%	10.34%	14.70%
5 im	12.91%	0.95	11.23%	15.03%
1 im w/augm.	13.38%	0.93%	11.52%	15.57%

Table 3: Mean Bit Error Rate comparison for auto-encoder binarization with different size of train base.

When we compare the results from Table 1 and Table 3, we note that the decoding with auto-encoder approach gives better results. We increase the decoding rate in average up to 91.25%. With classical thresholding the decoding rate is in average 65.35% and while using the decoding with feature set F3 and LDA, the decoding rate is in average 85.30%. That is why the use of neural network approaches can significantly increase the decoding rate of CSGC after P&S process.

5.3 Authentication test

In this section, we aim at study an impact of this estimation attack on the authentication test. The attack is as follows. An attacker scans the printed CSGC using the same scanner (as authentication center) and decodes the printed and scanned CSGC using either one of the tested classification methods (LDA with feature set F1 and F3) with 5 images in training base or an auto-encoder with $1 - 60$ images in training base. Then, s/he prints these counterfeited CSGC using the same printer.
The authentication center verify the authenticity of CSGC by:

(1) decoding the printed CSGC using:
   - classical thresholding with threshold $th = 127$,
   - Otsu binarization method,
(2) calculating the difference between the decoded CSGC and an original CSGC,

(3) comparing the obtained distance d with pre-calculated authentication threshold ϵ.

We use the BER as authentication test as it is easy to implement and it can give us an idea about the efficiency of our attack and estimation accuracy as was explained in Section 2.

The results of our decoding processes are presented in Table 4. We note that the best results are obtained for CSGC counterfeited using auto-encoder approach. The difference between the original and these counterfeited codes using thesholding and Otsu binarization for authentication is in average 3.05% and 2.78%, respectively.

Printed CSGC			BER	Std
Original	$\hat{I}_{127}$		34.65%	1.36%
	$\hat{I}_{otsu}$		**26.51%**	**1.01%**
Attack	$th = 127$	$\hat{I}'$	43.92%	1.09%
		$\hat{I}'_{F1}$	40.18%	1.21%
		$\hat{I}'_{F3}$	39.69%	1.05%
		$\hat{I}'_{60aug}$	37.10%	1.11%
		$\hat{I}'_{60}$	37.10%	1.22%
		$\hat{I}'_{5aug}$	37.41%	1.36%
		$\hat{I}'_{5}$	37.13%	1.33%
		$\hat{I}'_{1aug}$	**37.05%**	1.33%
	Otsu	$\hat{I}'$	35.14%	0.91%
		$\hat{I}'_{F1}$	35.32%	0.75%
		$\hat{I}'_{F3}$	34.41%	0.67%
		$\hat{I}'_{60aug}$	**29.29%**	0.81%
		$\hat{I}'_{60}$	29.65%	0.86%
		$\hat{I}'_{5aug}$	30.46%	0.93%
		$\hat{I}'_{5}$	30.75%	0.83%
		$\hat{I}'_{1aug}$	30.72%	0.94%

Table 4: Mean Bit Error Rate obtained while applying a classical thresholding with $th = 127$ and an Otsu's binarization for authentication test.

We plot the Receiver Operation Characteristics (ROC) curves for authentication using classical thresholding and Otsu's binarization in Fig. 4 and Fig. 5, respectively. By varying the authentication threshold ϵ, we compute the number of original (first print) and counterfeited (second print) codes that pass the authentication test. The percentage of accepted and rejected codes per attack are presented in Table 5. From this table we can conclude that if the opponent estimates the CSGC using classical thresholding, the counterfeited codes will not pass the authentication test. Let set the authentication threshold $\epsilon = 37.5$. With this authentication threshold ϵ all 100% original codes pass the authentication test. With the classification strategy using LDA with F3 feature set 5% of counterfeited codes can pass this authentication test. The results using auto-encoder approach are better: while using 60 images with data augmentation, 55% of codes can pass the authentication test. While using 1 images with data augmentation, 62.5% of codes can pass the authentication test with the same authentication threshold ϵ. We suppose that it is due to global binarization using the same threshold which is not adapted to this kind of images.

For authentication decoding using Otsu binarization, we have slightly different results as this binarizatoion threshold is adapted for each image.. The number of accepted and rejected codes per attack are

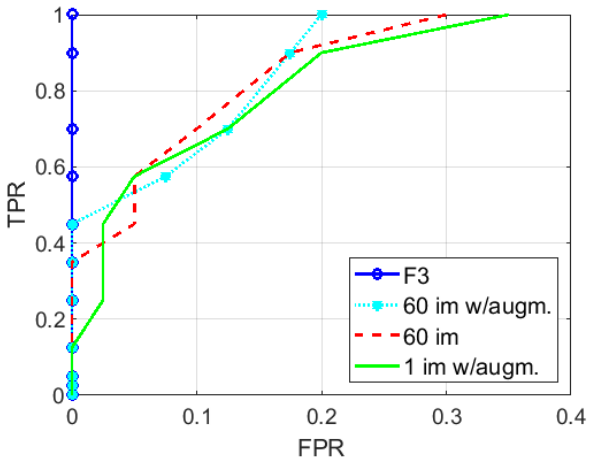


Figure 4: ROC curve for LDA with F3 characteristics vector and all auto-encoder decoding for authentication test that use $th = 127$ for CSGC decoding.

presented in Table 5. If we chose the authentication threshold equal to $\epsilon = 28.5$, all 100% of original codes can pass the authentication test, but also 17.5% of counterfeited codes using auto-encoder approach while using 60 images with data augmentation for training pass the authentication test. In comparison, there is no any image counterfeited using Otsu binarization or classification strategy using LDA with F3 feature set which can pass the authentication test with this ϵ.

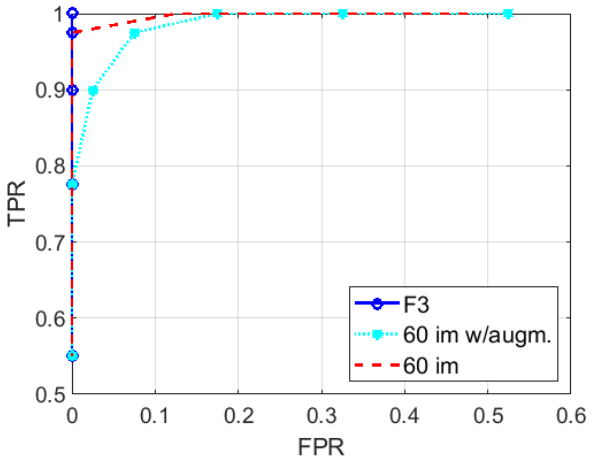


Figure 5: ROC curve for LDA with F3 characteristics vector and all auto-encoder decoding for authentication test that uses Otsu binarization for CSGC decoding.

The obtained results show us that the use of decoding techniques based on auto-encoders can help an attacker to create the counterfeited CSGC that will fool the authentication test.

	Original		Attack							
			same as authentication		LDA+F3		Auto-encoder 1 image w/aug		Auto-encoder 60 images w/aug	
	A	R	A	R	A	R	A	R	A	R
Authentication using tresholding with th = 127										
$\epsilon = 34.5$	45%	55%	0%	100%	0%	100%	2.5%	97.5%	0%	100%
$\epsilon = 35.5$	70%	30%	0%	100%	0%	100%	12.5%	87.5%	12.5%	87.5%
$\epsilon = 36.5$	**100%**	**0%**	**0%**	**100%**	**0%**	**100%**	**35%**	**65%**	**20%**	**80%**
$\epsilon = 37.5$	100%	0%	0%	100%	5%	95%	62.5%	37.5%	55%	45%
Authentication using Otsu's binarization										
$\epsilon = 26.5$	55%	45%	0%	100%	0%	100%	0%	100%	0%	100%
$\epsilon = 27.5$	90%	10%	0%	100%	0%	100%	0%	100%	2.5%	97.5%
$\epsilon = 28.5$	**100%**	**0%**	**0%**	**100%**	**0%**	**100%**	**0%**	**100%**	**17.5%**	**82.5%**
$\epsilon = 29.5$	100%	0%	0%	100%	0%	100%	7.5%	92.5%	52.5%	47.5%

Table 5: The percentage of accepted (A) and rejected (R) codes depending on authentication threshold ϵ for authentication using classical thresholding with $th = 127$ and using Otsu's binarization

6 CONCLUSIONS

In this paper we achieve a more efficient estimation attack using a supervised approach based on neural networks, here an auto-encoder. We obtain high decoding results while using quite small database (especially for neural network approaches). By training our neural network with 60 images of printed-and-scanned copy sensitive graphical code we obtain less then 10% as bit error rate. In future, we want to propose smarter authentication test and to test the proposed attacks with this smart authentication. The more realistic attacks need to be studied also. For example, the case when an attacker do not have the same printer-scanner as the authority center. In this case, an attacker needs to predict an impact of authentic P&S process.

REFERENCES

[1] G. Adams, S. Pollard, and S. Simske. 2011. A study of the interaction of paper substrates on printed forensic imaging. In *Proceedings of the 11th ACM symposium on Document engineering*. ACM, 263–266.

[2] C. Baras and F. Cayre. 2012. 2D bar-codes for authentication: A security approach. In *Signal Processing Conference (EUSIPCO), Proceedings of the 20th European*. 1760–1766.

[3] C. Baras and F. Cayre. 2013. Towards a realistic channel model for security analysis of authentication using graphical codes. In *Information Forensics and Security (WIFS), 2013 IEEE International Workshop on*. IEEE, 115–119.

[4] L. Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[5] J. Calvo-Zaragoza and A.-J. Gallego. 2019. A selectional auto-encoder approach for document image binarization. *Pattern Recognition* 86 (2019), 37–47.

[6] M. L Diong, P. Bas, C. Pelle, and W. Sawaya. 2012. Document authentication using 2D codes: Maximizing the decoding performance using statistical inference. In *IFIP International Conference on Communications and Multimedia Security*. Springer, 39–54.

[7] A. E. Dirik and B. Haas. 2012. Copy detection pattern-based document protection for variable media. *IET Image Processing* 6, 8 (2012), 1102–1113.

[8] X. Glorot and Y. Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

[9] R. N. Goldman. 1983. Non-counterfeitable document system. US Patent 4,423,415.

[10] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] A. T. P. Ho, B. A. M. Hoang, W. Sawaya, and P. Bas. 2014. Document Authentication Using Graphical Codes: Reliable Performance Analysis and Channel Optimization. *EURASIP Journal on Information Security* 2014, 1 (2014), 9.

[12] X. Mao, C. Shen, and Y-B. Yang. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*. 2802–2810.

[13] A-T. Phan Ho, B-A. Mai Hoang, W. Sawaya, and P. Bas. 2013. Document authentication using graphical codes: impacts of the channel model. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*. ACM, 87–94.

[14] J. Picard. 2004. Digital authentication with copy-detection patterns. In *Electronic Imaging 2004*. International Society for Optics and Photonics, 176–183.

[15] J. Picard and P. Landry. 2017. Two dimensional barcode and method of authentication of such barcode. US Patent 9,594,993.

[16] S. B Pollard, S. J Simske, and G. B Adams. 2010. Model based print signature profile extraction for forensic analysis of individual text glyphs. In *Information Forensics and Security (WIFS), 2010 IEEE International Workshop on*. IEEE, 1–6.

[17] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos. 2016. ICFHR2016 handwritten document image binarization contest (H-DIBCO 2016). In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 619–623.

[18] R. Schraml, L. Debiasi, and A. Uhl. 2018. Real or Fake: Mobile Device Drug Packaging Authentication. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 121–126.

[19] S. J Simske and G. Adams. 2010. High-resolution glyph-inspection based security system. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 1794–1797.

[20] O. Taran, S. Bonev, and S. Voloshynovskiy. 2019. Clonability of anti-counterfeiting printable graphical codes: a machine learning approach. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, United Kingdom.

[21] J. Tchan. 2004. The development of an image analysis system that can detect fraudulent alterations made to printed images. In *Optical Security and Counterfeit Deterrence Techniques V*, Vol. 5310. International Society for Optics and Photonics, 151–160.

[22] I. Tkachenko and C. Destruel. 2018. Exploitation of redundancy for pattern estimation of copy-sensitive two level QR code. In *Workshop on Information Forensics and security (WIFS), IEEE*.

[23] I. Tkachenko, C. Destruel, O. Strauss, and W. Puech. 2017. Sensitivity of different correlation measures to print-and-scan process. *Electronic Imaging* 2017 (2017).

[24] I. Tkachenko, W. Puech, C. Destruel, O. Strauss, J-M. Gaudin, and C. Guichard. 2016. Two-Level QR Code for Private Message Sharing and Document Authentication. *IEEE Transactions on Information Forensics and Security* 11, 3 (2016), 571–583.

[25] I. Tkachenko, W. Puech, O. Strauss, C. Destruel, and J-M. Gaudin. 2016. Printed document authentication using two level or code. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2149–2153.

[26] W. Wang, Y. Huang, Y. Wang, and L. Wang. 2014. Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 490–497.

[27] C-W. Wong and M. Wu. 2017. Counterfeit Detection Based on Unclonable Feature of Paper Using Mobile Camera. *IEEE Transactions on Information Forensics and Security* 12, 8 (2017), 1885–1899.