

CLONABILITY OF ANTI-COUNTERFEITING PRINTABLE GRAPHICAL CODES: A MACHINE LEARNING APPROACH

O. Taran, S. Bonev and S. Voloshynovskiy

University of Geneva, Department of Computer Science, Stochastic Information Processing Group
7, route de Drize, 1227 Geneva, Switzerland

ABSTRACT

In recent years, printable graphical codes have attracted a lot of attention enabling a link between the physical and digital worlds, which is of great interest for the IoT and brand protection applications. The security of printable codes in terms of their reproducibility by unauthorized parties or clonability is largely unexplored. In this paper, we try to investigate the clonability of printable graphical codes from a machine learning perspective. The proposed framework is based on a simple system composed of fully connected neural network layers. The results obtained on real codes printed by several printers demonstrate a possibility to accurately estimate digital codes from their printed counterparts in certain cases. This provides a new insight on scenarios, where printable graphical codes can be accurately cloned.

Index Terms— Printable graphical codes, clonability attack, machine learning.

1. INTRODUCTION

Counterfeiting of physical objects is a very important problem for the modern economies. There exist several techniques to protect original products against falsification and to provide a link between a physical object and its digital representation in centralized or distributed databases. This link can be implemented via *overt channels*, like personalized codes reproduced on products either directly or in a form of coded symbologies like 1D and 2D codes or *covert channels*, like invisible digital watermarks embedded in images or text or printed by special invisible inks. However, it is crucial to provide a non-clonability of this link to avoid any false acceptance of fake objects as authentic ones. Two most well known technologies that claim to ensure such a non-clonability are Physical Unclonable Functions (PUFs) [1] and Printable Graphic Codes (PGC) that originate from the work [2]. The theoretical comparison of PUFs and PGC is given in [3]. In this paper we focus on the clonability of PGC. The deployment of PGC has a lot of advantages and attracts many industrial players and governmental organizations. Nevertheless, the claimed

S. Voloshynovskiy is a corresponding author. The research was supported by the SNF project No. 200021_182063.

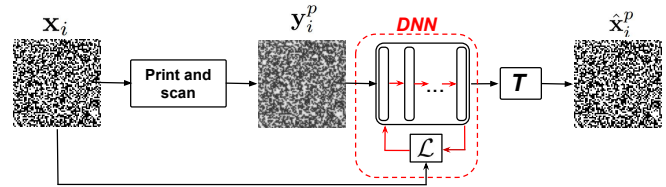


Fig. 1: Training procedure based on training samples (x_i, y_i^p) .

non-clonability of PGC remains largely unexplored besides some rare exceptions [4, 5].

The main goal of this paper is to investigate the resistance of PGC to clonability attacks. The overwhelming majority of such attacks can be split into two main groups: (a) *hand-crafted* attacks, which are based on the experience and know-how of the attackers and (b) *machine learning* based attacks, which use training data to create clones of the original codes.

In this paper, we focus on the investigation of *machine learning* based attacks due to the recent advent in the theory and practice of machine learning tools. Growing popularity and remarkable results of deep neural network (DNN) architectures in computer vision applications motivated us to investigate the clonability of PGC using these architectures trained for different classes of printers. In our study, we assume that the detection mechanism of defender is also unknown, thus making our attack universal in this sense.

Therefore, the main contributions of this paper are:

- we investigate the clonability of printable graphical codes using machine learning based attacks;
- we examine the proposed framework on real printed codes reproduced with 4 printers;
- we empirically demonstrate a possibility to sufficiently accurately clone the PGC from their printed counterparts in certain cases.

The reminder of this paper is organized as follows: Section 2 introduces a problem formulation and gives the details about the used test patterns, DNN architecture, training and test processes. Section 3 provides the details about the used printers, scanner and used dataset. The obtained empirical results and their analysis complete Section 3. Section 4 concludes the paper.




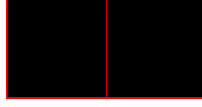



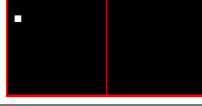
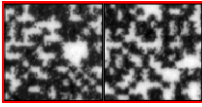


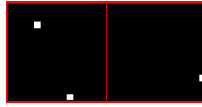
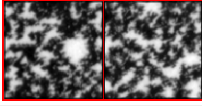


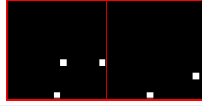
| | Printer | Scanned original | Original | Reconstructed (BN) | Difference |
|-----------------|---------|---|---|--|---|
| Laser printers | SA |  |  |  |  |
| | LX |  |  |  |  |
| Inkjet printers | HP |  |  |  |  |
| | CA |  |  |  |  |

Table 1: Examples of attacks against PGC: two samples of scanned codes, the estimates produced by *BN* model and the difference between the original and estimated codes.

2. PROBLEM FORMULATION

In our set up, we assume that the training dataset of the pairs of original (digital) $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\} \in \{0, 1\}^{N \times M}$ and printed codes $\mathbf{Y}^p = \{\mathbf{y}_1^p, \dots, \mathbf{y}_M^p\} \in \mathbb{R}^{N \times M}$ reproduced with popular high resolution printing technologies denoted as $p = \{1, \dots, P\}$ is given. We assume $\mathbf{y}_i^p \in \mathbb{R}^N$ due to the scanning of printed codes in general, or $\mathbf{y}_i^p \in \{0, \dots, 255\}^N$ in particular. The goal of the attacker is to obtain an accurate estimation of the original digital codes $\hat{\mathbf{X}}^p = \{\hat{\mathbf{x}}_1^p, \dots, \hat{\mathbf{x}}_M^p\} \in \{0, 1\}^{N \times M}$ for each printing technology p . Mathematically, it can be formulated as an estimate:

$$\hat{\mathbf{x}}_i^p = f_{\boldsymbol{\vartheta}^p}(\mathbf{y}_i^p), \quad (1)$$

where $i = 1, \dots, M$ and, in general case, $f_{\boldsymbol{\vartheta}^p}(\cdot)$ can be any trainable function with the parameters $\boldsymbol{\vartheta}^p$.

2.1. Deep neural network models

Nowadays, DNN technologies offer ample opportunities for training $f_{\boldsymbol{\vartheta}^p}(\cdot)$ in a form of parametrized deep architectures. We investigate the possibilities of two types of the DNN architectures for solving problem (1). The first one is based on several fully connected layers of the same size as the input data. Inspired by the fundamental role of autoencoders [6, 7] in unsupervised learning, we investigate a system based on a "bottleneck" structure, where the dimensionality of the layers is reduced to the middle layer and then expanded to the full dimensionality in the output layer.

For both architectures the general schema of the training procedure is shown in Fig. 1. For the given pairs of original and printed codes $(\mathbf{X}, \mathbf{Y}^p)$, it can be formulated as:

$$\hat{\boldsymbol{\theta}}^p = \arg \min_{\boldsymbol{\theta}^p} \sum_{i=1}^M \mathcal{L}(\mathbf{x}_i, \phi_{\boldsymbol{\theta}^p}(\mathbf{y}_i^p)) + \lambda \Omega_{\boldsymbol{\theta}^p}(\boldsymbol{\theta}^p), \quad (2)$$

where $\mathcal{L}(\cdot)$ is a loss function, $\phi_{\boldsymbol{\theta}^p}$ is a trained model, $\boldsymbol{\theta}^p$ represents the parameters of the trained model for chosen printer p and $\Omega_{\boldsymbol{\theta}^p}(\cdot)$ is a regularizer for the model parameters. In the case of an "bottleneck" model $\phi_{\boldsymbol{\theta}^p} = \phi_{\boldsymbol{\theta}_D^p}(\phi_{\boldsymbol{\theta}_E^p}(\cdot))$, $\boldsymbol{\theta}^p = (\boldsymbol{\theta}_E^p, \boldsymbol{\theta}_D^p)$ with $\boldsymbol{\theta}_E^p$ and $\boldsymbol{\theta}_D^p$ denoting the parameters of encoder and decoder parts, respectively.

In the vast majority of cases the original digital codes are binary. However, training the DNN model with binary output is not a trivial task due to the difficulties with derivatives and vanishing of the gradients. For this reason in our framework the output of the trained models is real. The binarization of the regenerated codes is performed via a simple thresholding with an optimal threshold estimated on the validation subset. Therefore, the function $f_{\boldsymbol{\vartheta}^p}$ in the equation (1) is:

$$f_{\boldsymbol{\vartheta}^p}(\cdot) = T_{t^p}(\phi_{\boldsymbol{\theta}^p}(\cdot)), \quad (3)$$

where $T(\cdot)$ is a thresholding function with the threshold parameter t^p and $\boldsymbol{\vartheta}^p = (\boldsymbol{\theta}^p, t^p)$.

At the test phase, the scanned sample \mathbf{y}_i^p is passed through the pre-trained DNN model. The estimation of the original code $\hat{\mathbf{x}}_i^p$ is obtained after thresholding T of the DNN output. The estimated code is printed and scanned on the corresponding equipment and the final decision about the code authenticity is made based on a chosen similarity measure $d(\cdot)$ between the original and printed estimated codes.

2.2. Test pattern

We used the *DataMatrix* symbology consisting of 72x72 modules, which is described in the international standard ISO/IEC 16022 [8]. To obtain a random bit distribution, the finder patterns were removed and only the mapping matrix of 64x64 modules was used for the printing tests.

It should be pointed out that although the *DataMatrix* code was initially proposed as an overt feature for personali-

sation applications, the chosen parameters of this code closely resemble those of the recently proposed *PGC* that might be equivalently printed up to a resolution of 2400 dpi. In our study we do not target to investigate the clonability of some particular *PGC*, but rather to demonstrate a general approach applicable to the majority of *PGC* designed with identical modulation principles.

3. EXPERIMENTS AND DISCUSSION

Digital printers. To evaluate the clonability aspects of *PGC* based on *DataMatrix* modulation and to investigate the influence of the printing technologies we use 4 digital printers: 2 inkjet printers HP OfficeJet Pro 8210 (*HP*) and Canon PIXMA iP7200 (*CA*) and 2 laser printers Lexmark CS310 (*LX*) and Samsung Xpress 430 (*SA*).

It should also be pointed out that up to our best knowledge there does not exist any accurate mathematical model describing the process of interaction between the ink and substrate (paper) besides some experimental studies as for example in [9, 5]. Due to the fact that in all our experiments we use the same paper, we skip this parameter in our models for simplicity, but the impact of the substrate can be investigated in a similar manner to the proposed one. Therefore, without loss of generality we assume that the model of printing process is unknown and is not required for our attack strategy.

DNN architectures. In our experiments we use two types of DNN architectures with the same input size equals to 576:

1. *FC*: fully connected DNN with 2, 3 and 4 hidden layers (hereafter referred to as *FC 2*, *FC 3* and *FC 4*). The size of each layer equals to the input size.

2. *BN*: "bottleneck" model with 2 fully connected hidden layers of size 256 and 128 at the encoder and decoder parts and a latent representation of size 36.

In both cases we used ℓ_2 norm as a loss function $\mathcal{L}(\cdot)$ in (2). The DNNs were implemented in Pytorch¹. The training of the models was done on the *Titan X* GPU card with the "Maxwell" architecture. All models were trained during 1 000 epochs with batch size equals to 128 and the learning rate equals $1e-3$.

PGC dataset. The dataset of 384 original binary codes \mathbf{X} of size 384×384 was generated according to the *DataMatrix* standard described in Section 2.2. To obtain the dataset of the printed/scanned codes \mathbf{Y}^p $p = \{1, \dots, P\}$, the original codes were printed on 4 printers ($P = 4$) and scanned with the Epson Perfection V850 Pro scanner at 1200 ppi. The pairs $(\mathbf{X}, \mathbf{Y}^p)$ were split into *training* (100 images), *validation* (50 images) and *test* (234 images) subsets. Taking into account that the input size of the used DNN models equals 576, each image was split into non-overlapping blocks of size 24×24 . Thus, the final size of the *training* set is 25 600 sub-images, the *validation* set contains 12 800 sub-images and the *test* set

| Method | SA | LX | HP | CA |
|------------------------------------|--------------|--------------|--------------|--------------|
| <i>Pearson correlation</i> | | | | |
| <i>Thr</i> | 0.774 | 0.766 | 0.742 | 0.704 |
| <i>FC 2</i> | 0.995 | 0.994 | 0.982 | 0.981 |
| <i>FC 3</i> | 0.994 | 0.994 | 0.982 | 0.983 |
| <i>FC 4</i> | 0.994 | 0.995 | 0.981 | 0.982 |
| <i>BN</i> | 0.996 | 0.996 | 0.986 | 0.984 |
| <i>normalized Hamming distance</i> | | | | |
| <i>Thr</i> | 11 | 12 | 13 | 15 |
| <i>FC 2</i> | 0.22 | 0.24 | 0.93 | 0.98 |
| <i>FC 3</i> | 0.23 | 0.24 | 0.90 | 0.85 |
| <i>FC 4</i> | 0.24 | 0.23 | 0.95 | 0.90 |
| <i>BN</i> | 0.21 | 0.22 | 0.69 | 0.76 |

Table 2: Regeneration accuracy with respect to original codes consists of 59 904 sub-images.

It should be pointed out that the training procedure is blind in the sense that we did not use any information about the principles of the *DataMatrix* code generation.

To evaluate the accuracy of the prediction of "regenerated" codes we use *Pearson correlation* and normalized *Hamming distance* between the original digital codes and the corresponding regenerated ones. The obtained results are presented in Table 2. Additionally to the DNN models, we perform the estimation from the printed codes via a simple thresholding (without DNN processing) similarly to [4, 5, 2]. The obtained results correspond to the *Thr* method in Table 2 and serve as baseline error. From the presented results, it is clear that the *BN* architecture provides the best results. To provide more understanding how the codes look, we visualize the sub-blocks of size 84×84 from several codes for each printer and the estimations deploying the *BN* as the best estimator in Table 1.

To answer the question if the amount of errors in the *BN* regenerated codes can be noticed by the defender and how the *BN* results differ from the baseline estimation obtained via *Thr* method, we printed our estimated codes for both cases on the same printers with the same parameters as the original codes and after that we scanned them on the same scanner. To evaluate the authenticity of the obtained results a number of metrics can be used. However, according to the authors in [5] the most used one is a comparison of an original with a binarized or grey level version of the printed code. The authors in [5] claim that the comparison with the grey level observations is preferable, since binarization is a lossy transformation. In our evaluation we use *Pearson correlation* between the originals and grey level printed codes. Additionally, we use normalized *Hamming distance* to measure the accuracy of the logical symbol estimation in the originals and binarized printed codes. Using these statistics, we compute the ROC curves based on the probability of correct detection P_d

¹<https://github.com/taranO/clonability-of-printable-graphical-codes>

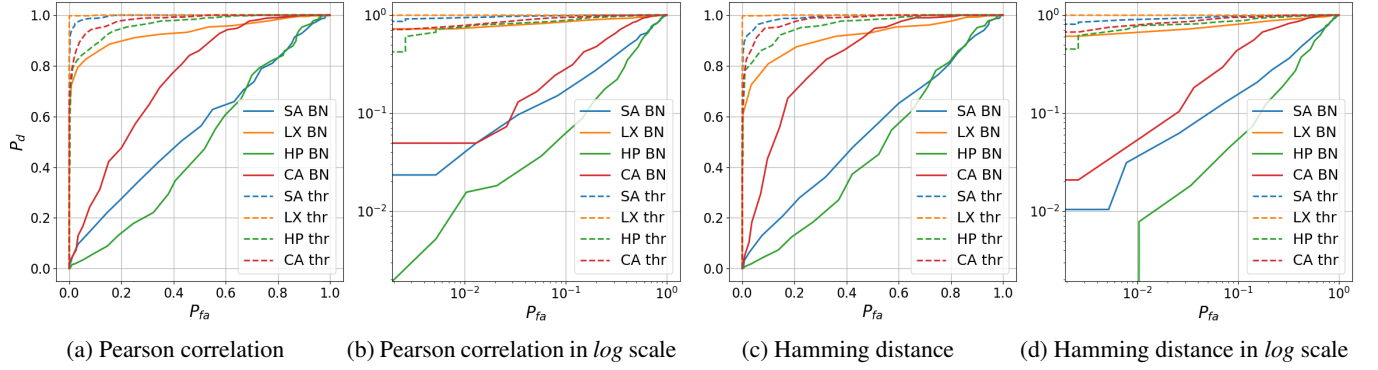


Fig. 2: The ROC curves for *Pearson correlation* and *Hamming distance* between the original and fake printed codes estimated via *BN* and *Thr* methods. P_d denotes to the probability of the correct detection and P_{fa} is the probability of false acceptance.

and the probability of false acceptance P_{fa} via:

$$\begin{aligned} P_d &= \Pr\{\alpha \cdot d(\mathbf{x}_i, \mathbf{y}_i^p) \geq \gamma | \mathcal{H}_0\} \\ P_{fa} &= \Pr\{\alpha \cdot d(\mathbf{x}_i, \mathbf{y}_i^p) > \gamma | \mathcal{H}_1\}, \end{aligned} \quad (4)$$

where γ is the threshold, $d(\cdot)$ is a similarity measure between the original and printed codes, \mathcal{H}_0 corresponds to the hypothesis that \mathbf{y}_i^p is an authentic code and \mathcal{H}_1 is the hypothesis that \mathbf{y}_i^p is a fake (cloned) code, α equals to 1 for the *Pearson correlation* and to -1 for *Hamming distance*.

The obtained ROC curves are illustrated in Fig. 2. It is easy to see that comparing with the baseline estimation via *Thr* method, the system with *BN* models makes the fake detection more difficult for defender. Particularly, as it can be noticed from Fig. 2, in contrast to *Thr* based estimation in the case of *SA* and *HP* printers, it is absolutely impossible to reliably distinguish the originals and fakes. For the *SA* printer this result is evident due to the previously demonstrated high quality estimation. In the case of *HP* printer such a result can be explained by the fact that, besides a quite big amount of errors in the estimated codes, the printing quality is relatively poor due to the high dot-gain. This leads to a sufficient amount of errors in the original printed codes that are masked in the printed fake codes due to the dot-gain effect. As a result, both codes become very close. In the case of the *CA* printer, the ROC behaviour is superior, which is expected due to the previously demonstrated low quality estimation. The situation with the *LX* printer is the most interesting. From one point, the printing quality of this printer is a little bit worse than for the *SA* printer and the obtained estimated error is not much higher. However, detailed analysis shows that the distributions of the errors between the codes is different. In the case of the *SA* printer there are about 50% of estimated codes without any mistake. This makes these fakes undistinguishable for the detector and the general quality of fake detection low. In the case of the *LX* printer, the errors are distributed more or less uniformly between the codes, in the sense that almost each code has estimation errors. Due to the high printing quality this makes these codes "better"

distinguishable for the detector. Nevertheless, it should be pointed out that the general level of false acceptance for the *LX* printer is too high for practical use. For example, as can be seen from Fig. 2 for the probability of correct detection of around 0.95 the probability of false acceptance is 0.6. To have the P_{fa} close to 0, one can achieve the P_d of only about 0.6 - 0.8. For practical applications P_d should not be less than 0.99 with the P_{fa} not exceeding 10^{-6} . From this we can conclude that the obtained results demonstrate the low resistance of the PGC based on *DataMatrix* modulation and similar codes to the *machine learning* based clonability attacks.

4. CONCLUSIONS

In this paper we investigated the clonability of printable graphical codes using *DataMatrix* modulation typical for many *PGC* designs using *machine learning* based attacks. We tested the proposed framework with two different DNN architectures on real printed data. We empirically proved the possibility to accurately estimate the printable codes for high quality printers even from the relatively small training datasets. Based on the performed experiments and obtained results we can identify three main criteria for successful fake detection: (a) the printing quality, (b) the amount of errors in estimated codes and (c) the regularity of the estimated errors. The defenders should prefer average quality printers with a dot-gain sufficient to make regular errors in the originals estimation. Moreover, the results show that modern machine learning technologies make the printable graphical codes vulnerable to clonability attacks.

For future work, we aim at examining other types of graphical codes, at investigating the possibilities of mobile phones for the detection of fake codes and to compare the abilities of machine learning approaches versus hand-crafted attacks. Finally, we plan to consider *GAN*-like architectures to produce even more accurate fakes. The impact of the number of training examples and training from the original digital templates are also amongst our future priorities.

5. REFERENCES

- [1] Sviatoslav Voloshynovskiy, Oleksiy Koval, Fokko Beekhof, and Thierry Pun, “Unclonable identification and authentication based on reference list decoding,” in *Proceedings of the conference on Secure Component and System Identification*, Berlin, Germany, March 17–18 2008.
- [2] Justin Picard, “Digital authentication with copy-detection patterns,” in *Optical Security and Counterfeit Deterrence Techniques V*. International Society for Optics and Photonics, 2004, vol. 5310, pp. 176–184.
- [3] S Voloshynovskiy, P Bas, and T Holotyak, “Physical object authentication: detection-theoretic comparison of natural and artificial randomness,” in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, Shanghai, China, March, 20–25 2016.
- [4] Cléo Baras and François Cayre, “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, 2013, pp. 115–119.
- [5] Anh Thu Phan Ho, Bao An Mai Hoang, Wadih Sawaya, and Patrick Bas, “Document authentication using graphical codes: impacts of the channel model,” in *Proceedings of the first ACM workshop on Information hiding and multimedia security*. ACM, 2013, pp. 87–94.
- [6] Geoffrey E Hinton and Ruslan R Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [8] *ISO/IEC 16022: Information technology - Automatic identification and data capture techniques - Data Matrix bar code symbology specification*, 2006.
- [9] Renato Villán, Sviatoslav Voloshynovskiy, Oleksiy Koval, and Thierry Pun, “Multilevel 2d bar codes: Towards high capacity storage modules for multimedia security and management,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, pp. 405–420, December 2006.