A Two-Step Method for Ensuring Printed Document Integrity using Crossing Number Distances

Felix Yriarte, Pauline Puteaux, Iuliia Tkachenko

A Two-Step Method for Ensuring Printed Document Integrity using Crossing Number Distances

Felix Yriarte[†⋆], Pauline Puteaux[⋆] and Iuliia Tkachenko[†]
[†] *Univ Lyon, Univ Lyon 2, CNRS, INSA Lyon, UCBL, LIRIS, UMR 5205, F-69676 Bron, France*
[⋆] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France*
felix.yriarte@liris.cnrs.fr, pauline.puteaux@cnrs.fr, iuliia.tkachenko@liris.cnrs.fr

***Abstract*—Nowadays, with the use of photo-editing software being mainstream, document integrity verification has become crucial. As we have seen during the pandemic, most administrative documents are printed and then scanned before being transmitted, making these documents noisy. Indeed, a printed and scanned document undergoes geometric transformations, as well as the addition of black spots, not to mention a decrease in color intensity. The relevant features of an original document, which will be matched against a query document, are stored to be used as a template. We propose a 2-step method that compares a template with a query document to ensure that the query document has not been tampered with. Our method first reverts geometric transformations the document underwent, and then extracts the crossing numbers in that image. A Euclidean distance based matching method is applied to the two sets of crossing numbers, and abnormally distant point groups are flagged as potentially modified. A second step in our method is then applied to analyze the statistical properties of these distance values, to ensure that the document has not been altered. Our results when we apply our method to a database containing administrative documents and tampered versions of these documents – all of which underwent a print and scan process – show the validity of our considerations.**

***Index Terms*—document integrity check, print-and-scan process, printed document, document forgery.**

I. Introduction

Although a number of administrative documents are still distributed in paper form, most of these are then transmitted over the internet, in digital form. Verifying the integrity of digital documents is not a simple task, particularly because of how easy it is to use powerful image editing tools (like Photoshop or GIMP). Additionally, deep learning technologies have recently been used to produce high quality document forgeries [14].

Document hashing works well for digital documents, for example by using OCR (Optical Character Recognition) techniques paired with cryptographic hashing functions [13]. However, OCR stability and accuracy significantly drop if the document is printed and scanned once [1], and plummet after a double print-and-scan (P&S) process [12].

Printed document integrity check can also be done using printer forensics techniques. In this approach, the specific features such as noise intensity, contour roughness, and average gradient of character edges are analyzed as to identify the forged areas [6]. The main drawback of such approaches is the necessity of having knowledge about the printer and scanner used.

Document forgery detection can also be done by constructing document-specific hashes. In this approach, features are extracted from each character, then encoded so that the resulting codes can be used for integrity check [11]. Several of these features – which are also used in biometrics – were shown to be robust to the P&S process [5].

In this work, we deal with administrative document falsification. A genuine document is generated by the authorities. The document's signature is computed and stored in the authority's database. To compute that signature, the coordinates of the features considered in [5] are extracted. That signature can also be stored in a barcode, that could even be integrated to the document. The document could still be used either in digital form, or as a printed and scanned document. When the user transmits the document to any entity, its integrity can be verified by comparing its signature with the stored signature. An overview of the studied document life cycle is illustrated in Fig. 1, both for a genuine and an attack scenario.

The rest of the paper is organized as follows. We present the proposed document signature extraction system in Section II and document integrity check in Section III. The experimental results are presented in Section IV. Finally, we conclude this work and discuss future prospects in Section V.

II. Document signature extraction

We propose a method for document integrity verification that is robust to the P&S process. This method consists of a signature extraction step and of a verification step. The proposed integrity check system is illustrated in Fig. 2. The signature extraction step consists of 1) document image preprocessing and 2) character feature extraction. The integrity check step is applied to identify fields that present abnormal features, and then to confirm or deny that they are indeed forged.

In this section we overview the necessary pre-processing operations and present the features used. The proposed feature matching method is described in Section III.

A. *Pre-processing operations*

The P&S process adds different types of degradations to a document: geometric transformations, addition of black spots, gray-level value change (P&S images are grayscale, whereas digital images usually are binary), as well as compression artifacts that appear after the scan [9]. In order to eliminate

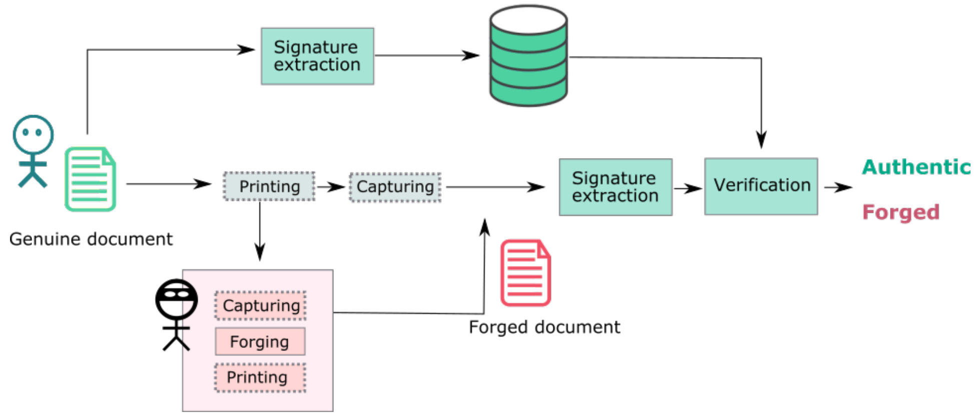


Fig. 1. Overview of the considered scenario: the green blocks correspond to authorities, the red blocks correspond to a forger. The dashed parts represent optional processes.

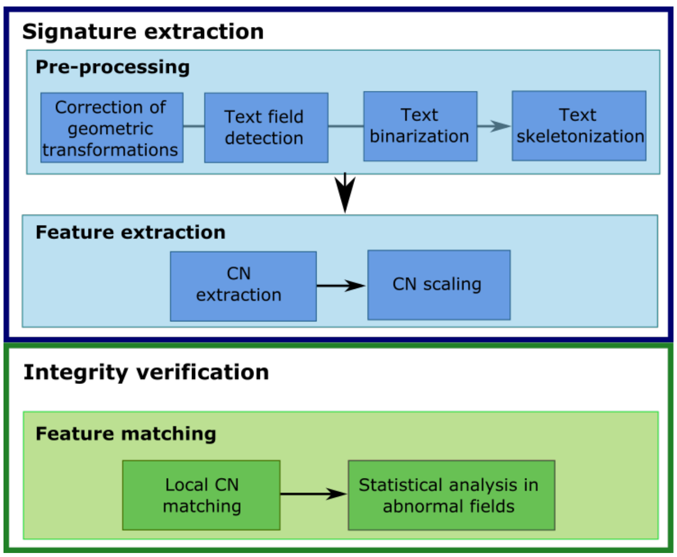


Fig. 2. Pipeline of the document image pre-processing steps and text integrity verification.

part of the deterioration that is due to the P&S process, several pre-processing operations are applied to a document image. Here we list the pre-processing operations that are shown in our document integrity check pipeline illustrated in Fig. 2.

- **Correction of geometric transformations.** Scanning a printed document adds different geometric transformations to the document image, such as rotation, translation, scale change, as well as crop. Even if the scanning operation is done carefully, the resulting image is bound to undergo geometric transformations. We set up a transformation correction process that is particularly adapted to the database we consider – that is composed of payslips (see Fig. 5.a-b): translations are corrected by cropping the document image so that only its frame and content remain. Using the identified position of the frame's corners, rotations are estimated, and corrected. Rather than correcting the scale operations on the image itself, we scale the coordinates of the extracted feature points so as not to incur interpolation errors during the image scale operation.
- **Text content detection.** We then apply a text detection step so as to remove spots, and non-textual image contents, such as tables, or the document frame. Indeed, after a P&S process, it is common to find small black spots on the document image. These spots usually are smaller than a character.
  We apply the edge-detection method proposed by Suzuki and Keiichi [10], and filter out unusually small or large content. This allows us to shape an image mask – a binary image of the same resolution as the document image – indicating the fields that are to be considered.
- **Binarization and skeletonization.** In order to extract the feature points from a document image, it is necessary to apply a skeletonization step to it. A skeletonization is a shape thinning operation that returns a 1 pixel-wide skeleton that preserves the connectivity of components. We use the method proposed by Lee in [2]. Before the skeletonization, the image is thresholded using the thresholding method proposed by Otsu [3].

After the application of these pre-processing operations, a skeleton image is obtained, and features can be extracted.

B. Feature extraction

Since the P&S process induces noise, it is important to take into account features that are robust to such operations. Crossing Numbers (CN) are used in biometrics (namely, fingerprint recognition) as features because of their stability through acquisition noise. They can however also be used to identify textual characters [5], [11]. A skeleton pixel's crossing number value is the number of neighbouring skeleton pixel it has. This gives an indication on the connectivity of every skeleton pixel. The different CN values that can be encountered are shown in Fig. 3. We do not consider CN values of 2 for our matching step, since they cannot be used to distinguish different characters.

Furthermore, some characters can present serifs – small lines or strokes attached to the end of longer strokes. Serifs cause the addition of two types of crossing numbers: a bifurcation and up to two ending points. At lower resolutions, serifs are not reliably identified and can be missed ; in which case the matching step fails for CN feature points extracted from serifs. Since these features are not significant for the matching step, we remove them from the feature point set during the CN feature point extraction.

CN = 0	CN = 1	CN = 2	CN = 3	CN = 4
Isolated point	Ending point	Connective point	Bifurcation point	Crossing point

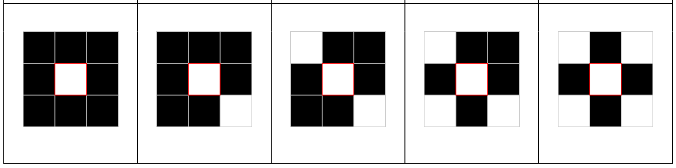


Fig. 3. Five pixel types as a function of their associated crossing number (the centered pixel framed in red is the pixel of interest).

Depending on the scanning quality used, the resulting P&S image's resolution differs. It is thus necessary to scale the extracted CN feature points so as to normalize the data. However, we do not want to induce interpolation errors by applying a scaling operation to the image, which could add discontinuities to a skeleton image. That is why we scale the coordinates of the extracted CN feature points, rather than scaling the document image itself. To do so, every extracted CN feature point's coordinates are multiplied by the ratio $\frac{\text{Reference\_Image\_Resolution}}{\text{Query\_Image\_Resolution}}$, as shown in Algorithm 1, lines 2-7. Note that every reference document – since they are numeric – has the same image resolution. We consider that resolution known when scaling the extracted CN feature points.

III. Document integrity check

After the CN feature points are extracted from the query document image, they are matched against the reference document image feature points (stored in a database as a signature), so as to verify the integrity of the query image. In this section, we introduce the two-step document integrity verification method we propose.

The first step of this method identifies abnormal fields – considered as possibly forged – for which no good candidate features are found in the reference. The second step of our method determines whether these abnormal fields are falsified, or highly distorted only.

A. Local CN matching

Document falsifications often consist of minor changes (like name, surname, or a salary amount). Therefore, global integrity check approaches do not work well.

Algorithm 1 presents the first step of our integrity check method. A query document is taken as an input. It can have gone through a P&S process, and can either be genuine or falsified.

After the CN feature points are extracted from the query document image, they are re-scaled and are then matched against the signature of an authentic digital document considered as a reference. For every reference CN feature point, the closest query CN feature point of any type is found, and the distance between the two points is stored.

We then consider, for every field that was highlighted in our image mask, the reference CN feature points belonging to that field. This allows us to consider local values, rather than global statistical properties regarding distances. For every field, we can plot the average matching distance and observe if some distances are particularly high.

Algorithm 1: Local CN matching

Require: **Reference_CN_Set**, **Query_Image**, **Reference_Image_Resolution**, **Query_Image_Resolution**

Ensure: **Distances** - Corresponding matching distance for each reference CN

1: **Struct** CN {posX, posY, value};
2: Scale_X ← Reference_Image_Resolution.**X** / Query_Image_Resolution.**X**;
3: Scale_Y ← Reference_Image_Resolution.**Y** / Query_Image_Resolution.**Y**;
4: Query_CN_Set ← CN_Extraction(Query_Image);
5: **for** cn ∈ Query_CN_Set **do**
6:     cn ← scale(cn, Scale_X, Scale_Y);
7: **end for**
8: **List Distances**;
9: **for** cn ∈ Reference_CN_Set **do**
10:     Distances ← Distances ∪ $\{\min_{p \in \text{Query\_CN\_Set}} \text{dist(cn,p)}\}$;
11: **end for**

Fields for which the matching distance is abnormally high are more likely to contain falsifications. However, it is also possible that they are only subject to geometric transformations. These transformations are not always perfectly corrected during the pre-processing step, and can cause slightly increased matching distances for some fields – mostly the document's borders. For example, as shown in Fig. 4.a, the matched sequences are identical, yet the matching distance for each CN feature point is larger than zero. Because of that, some fields can show higher than average matching distances, although they do not contain altered characters. However, we propose a second step where the distribution of matching distances in a field is analyzed, so as to differentiate falsified character sequences and characters that underwent heavy geometric transformations.

B. Statistical analysis in abnormal fields

We formulate the following hypothesis: if two identical but shifted character sequences are matched against each other, the matching distances are almost similar, as illustrated in Fig. 4.a. The histogram representing the distribution of these matching distances shows low value dispersion (Fig. 4.b). However, when matching against a falsified sequence, the matching distances are relatively random (Fig. 4.c-d), as long as the

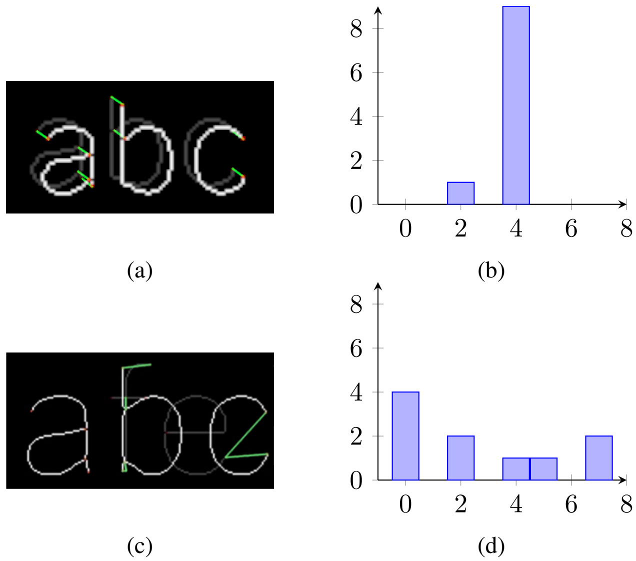


Fig. 4. Example of a) identical shifted sequences of characters (authentic document), b) histogram of distances associated to (a), which illustrates weak dispersion, c) different sequences of characters (forged document), d) histogram of distances associated to (c), which illustrates high dispersion.

distributions are analyzed locally – considering the matching distances of the CN feature points inside a field, for instance. If the matching distance distribution within a field is uneven, it is unlikely to contain falsified characters. We can not however say that a distribution that is close to uniform necessarily is falsified, which is why we first locate fields containing high matching distances, and then check whether their distance distribution is uniform or not. A field that both contains high matching distances and of which the distribution is close to uniform is likely to be falsified. If it contains high matching distances of which the distribution is uneven however, it is not likely to contain altered character sequences. To evaluate the distribution of distances within a field, we use the Shannon entropy [7] value of the matching distances $H(D)$, so as to get an idea on whether a distribution is uneven or close to uniform:

$$H(D) = -\sum_{i=0}^{l-1} p(d_i)\log_2(p(d_i)), \quad (1)$$

where D is the set containing all the distances within a field, of size k, and $p(d_i)$ is the probability of occurrence of a distance value d_i ($0 \leq d_i < l$).

One can note that the highest distance value $l-1$ depends on the document used. In order to have normalized entropy values (between 0 and 1 bit), we then divide $H(D)$ by the maximal entropy value, which is equal to $\log_2(\min(k,l))$ [4].

IV. Experimental results

In this section, we first describe the document database we have considered to perform our experimental results (Section IV-A). Then, in Section IV-B, we present the obtained results by applying our proposed document integrity check method to this database. Finally, in Section IV-C, we provide a discussion on the limitations and the drawbacks of our method and give some directions that can be investigated in future work.

A. Document database used

The Payslip database was proposed by Sidere *et al.* in [8]. It is composed of 200 digitally generated payslips, using names, first names, and addresses among the most common in France. The information presented in these bulletins is therefore fictitious (it is not supposed to represent real people), but close to reality (the different values of the fields are taken from the real administrative documents). A genuine and a forged document are illustrated in Fig. 5.a-b.

The database also contains several falsified versions of every document. The database is made up of 200 genuine documents, and 477 falsified documents. These falsifications were however carried out on digital documents, or on documents that were very slightly altered by a P&S process. We therefore printed and scanned a subset of these documents (both genuine and altered versions) in order to verify the robustness of our method to the P&S process.

We have considered documents with Arial font and a font size of 10 only. The choice of these font type and size are random. We believe that the proposed integrity check works for any font type and size. The subset of image documents used for our experiments[1] is detailed in Table I.

For our experiments, we use 10 different numeric documents as reference documents. We only consider query documents that have gone through a P&S process, as the first step of our proposed method is sufficient to verify the integrity of a numeric document.

	Genuine	Forged
P&S 300 dpi	10	21
P&S 600 dpi	10	21
Double P&S 600 dpi	10	21

TABLE I
Description of the database used for our experiments.

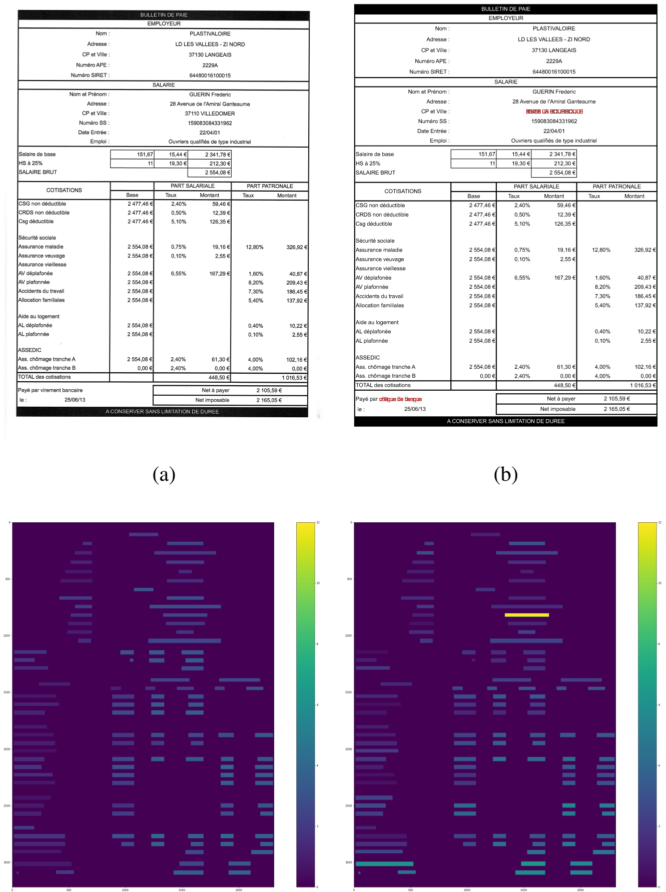
The implementation of our method was done using Python and standard image processing libraries, such as OpenCV, matplotlib, and scikit-image[1].

B. Integrity check results

In Fig. 5, we provide an example when applying our proposed method on both a genuine and a forged versions of a document of the database, considering a P&S resolution of 600 dpi (dots per inch). The genuine version is depicted in Fig. 5.a, while the forged version is shown in Fig. 5.b. For better readability, the falsifications are framed in red. One can see that two fields are forged: the postal zip code and the city name (top of the document) have been modified, as well as the payslip mean of payment (bottom of the document). Therefore, these falsifications occur on both letters and numbers.

In Fig. 5.c and Fig. 5.d, we have illustrated the first step of our method by showing the distance maps associated to the

[1]To stimulate a collaboration and reproducible results, the python code as well as the augmented Payslip database are publicly available via this link: https://gitlab.liris.cnrs.fr/gdr_isis_fuzzydoc/fuzzydoc_cn_distances



(a) (b)

(c) (d)

Fig. 5. An example of a) genuine document, b) forged document with highlighted forged parts, c) map of genuine document, d) map of forged document.

genuine and the forged documents, when comparing them with the associated numeric template document. In order to generate these distance maps, we computed the average distance values for each field of the documents. In Fig. 5.c, the average distance value is smaller than 5 pixels in every fields. Distance values are always small because the document is authentic: the offsets when comparing some CN feature points' coordinates between the template and the query document are only due to the noise during the P&S process. Conversely, in Fig. 5.d, we can see that the two forged fields stand out. Indeed, the average distance values in these fields are equal to 12 pixels and 7 pixels. However, one can note that, in the bottom of this map, some other fields that are not forged are highlighted.

The second step of our method is then carried out in order to analyse the dispersion of the distance values in all the fields which are identified as possibly forged due to the fact that there are high distance values. We thus perform a Shannon entropy measurement. The measured entropy values in the two forged fields are equal to 0.69 bit and 0.81 bit (after a normalisation between 0 and 1). These values indicate that the distance distributions associated to the two fields are quite uniform. This characterises the presence of falsifications. When we compute Shannon entropy of the fields misidentified as possibly forged at the bottom of the Fig. 5.d, the values are quite small (*i.e.* smaller than 0.5 bit), because the distance values are similar in these fields despite them being high (as explained in Section III-B).

We have then applied our proposed method on the whole database described in Section IV-A. In order to evaluate the accuracy of the proposed method for the two-class classification task (genuine *vs* forged document), we have defined different thresholds. Note that these thresholds do not depend on the P&S resolution because we suppose that, in a real life scenario, we have little to no information on the P&S parameters.

For the first step of our method, a document is considered as possibly forged if there exists at least one field in the document containing two matching distance values greater than 12 pixels. Indeed, a falsification occurs when at least one character is modified in the document, and there are at least two CN feature points in a character (except for 'o' and '0', which have no CN $\neq 2$). Moreover, a distance between two CN feature points – one from the template and the other from a P&S document – is considered as large if it is greater than 12 pixels. According to our experiments on various documents, this is a good choice to avoid the misdetection of a forged document, while limiting the number of genuine documents classified as forged.

For the second step of our method, we perform Shannon entropy measurements only on the fields that are identified as possibly forged, according to the first step. This measurement gives us an indication on the dispersion of the distance values in a field. We consider that the distance distribution is uniform as long as the measured entropy value is greater than 0.5 bit (after a normalisation between 0 and 1).

In Table II, we present the accuracy of the proposed method for the two-class classification task as a function of the considered P&S resolution, using the thresholds we have just presented. Under the assumption that we cannot guess the P&S resolution, we can see that the genuine documents are correctly identified in 83% of the cases, while the forged documents are correctly classified in 95% of the cases. These results are pretty interesting and show the relevance of our proposed method of document integrity check.

However, it can be seen that at double P&S600, only 50% of the genuine documents are classified correctly. We believe that by adapting our pre-processing step and threshold values to that particular resolution, we would obtain better results. From a security point-of-view, it is crucial that little to no forged documents get classified as genuine, even if that means sacrificing part of the accuracy for double P&S600 genuine documents.

In order to perform a deeper analysis on the failure cases using our method, we provide a discussion on its limitations and drawbacks in Section IV-C.

	Genuine	Forged
P&S 300 dpi	100%	95%
P&S 600 dpi	100%	90%
Double P&S 600 dpi	50%	100%
Total	**83%**	**95%**

TABLE II
ACCURACY OF THE PROPOSED METHOD FOR THE TWO-CLASS CLASSIFICATION TASK (GENUINE *vs* FORGED DOCUMENT) AS A FUNCTION OF THE CONSIDERED P&S RESOLUTION.

C. Discussion

Our proposed method is very efficient when considering relatively large falsifications – that make up the majority of a field – but no so efficient for falsifications that only alter one or two characters within a field. Because we consider a statistical analysis of matching distance distributions, a distribution that contains a small amount of points, or a small proportion of falsified points is harder to classify.

We have also noticed that falsifications concerning the ‘o’ and ‘0’ characters were usually not detected, since these characters contain no CN$\neq$ 2. Because we match reference CN feature points to query CN feature points, no forgery will be detected if there is no CN feature point to match. If a ‘0’ character is changed into a ‘9’ by an attacker, the falsification can not be identified. Similarly, a falsification that consists of an addition of characters to a field will not be detected, since no reference feature points will match against the altered characters. It can be seen in Table II that our proposed method achieves an accuracy of about 92% for P&S300 and P&S600 resolutions when dealing with forged documents. The few forged documents that were incorrectly classified as genuine contained character additions and falsifications that only represent a small proportion of the field. We believe that part of the issue can be fixed by also considering the matching between the query CN feature point set as reference, and the reference CN feature points as query. This would identify additions, but would also increase the impact of P&S noise.

A possible solution against the non-detection of ‘o’ and ‘0’ characters is the augmentation of the CN feature points, by taking into account the loops in a document, for instance. It is also to be noted that because we have worked under the assumption that the P&S resolution was not known when applying our proposed method to a P&S document, we had to set up thresholds that would be efficient for every P&S resolution. However, if we were to identify the P&S resolution when considering a query document, we could use relevant threshold values for every possible resolution.

V. Conclusion

In this paper, we propose an efficient method for document integrity verification, that is robust to the print-and-scan process. As a genuine document is created and transmitted, its signature is computed and stored in a database: the document is pre-processed, and its crossing numbers are stored, making up that signature. If a possibly printed-then-scanned document poses as the first one, it is pre-processed, its CN feature points are extracted, and then matched against the stored signature. We consider the Euclidian distance between CN feature points’ coordinates, and then their local distribution, so as to identify potential forgeries. Our proposed method is able to both identify a forged document, and locate the field containing the falsification. Our method is particularly efficient when considering documents that have only been printed-and-scanned once, with a resolution of either 300 or 600 dpi.

In future work, we would like to explore the use of feature augmentation, so as to identify falsifications operated on characters that contain no CN feature points. The identification of the P&S resolution used for a query document would also be an interesting track to investigate, as it would allow for the consideration of different pre-processing methods and thresholds, depending on the P&S resolution used.

Acknowledgment

This work was funded by the project *FuzzyDoc* supported by the CNRS Research Group of Information, Signal, Image and Vision (CNRS GdR-ISIS).

References

[1] S. Eskenazi, P. Gomez-Krämer, and J-M. Ogier. A study of the factors influencing ocr stability for hybrid security. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 9, pages 3–8. IEEE, 2017.

[2] C. C. Lee. A Sequential Thinning Algorithm For Image Skeletonization. In Andrew G. Tescher, editor, *Applications of Digital Image Processing VI*, volume 0432, pages 287 – 291. International Society for Optics and Photonics, SPIE, 1984.

[3] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[4] P. Puteaux and W. Puech. Noisy encrypted image correction based on shannon entropy measurement in pixel blocks of very small size. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 161–165. IEEE, 2018.

[5] P. Puteaux and I. Tkachenko. Crossing number features: from biometrics to printed character matching. In *International Conference on Document Analysis and Recognition*, pages 437–450. Springer, 2021.

[6] S. Shang, N. Memon, and X. Kong. Detecting documents forged by printing and copying. *EURASIP Journal on Advances in Signal Processing*, 2014(1):1–13, 2014.

[7] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[8] N. Sidere, F. Cruz, M. Coustaty, and J-M. Ogier. A dataset for forgery detection and spotting in document images. In *2017 Seventh International Conference on Emerging Security Technologies (EST)*, pages 26–31. IEEE, 2017.

[9] K. Solanki, U. Madhow, BS. Manjunath, S. Chandrasekaran, and I. El-Khalil. Print and scan resilient data hiding in images. *IEEE Transactions on Information Forensics and Security*, 1(4):464–478, 2006.

[10] S. Suzuki and Keiichi A. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[11] L. Tan and X. Sun. Robust text hashing for content-based document authentication. *Information Technology Journal*, 10(8):1608–1613, 2011.

[12] I. Tkachenko and P. Gomez-Krämer. Robustness of character recognition techniques to double print-and-scan process. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 9, pages 27–32. IEEE, 2017.

[13] R. Villán, S. Voloshynovskiy, O. Koval, F. Deguillaume, and T. Pun. Tamper-proofing of electronic and printed text documents via robust hashing and data-hiding. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 633–644. SPIE, 2007.

[14] L. Zhao, C. Chen, and J. Huang. Deep learning-based forgery attack on document images. *IEEE Transactions on Image Processing*, 30:7964–7979, 2021.