Forensic Science International 247 (2015) 105–125

Contents lists available at ScienceDirect

Forensic Science International

journal homepage: www.elsevier.com/locate/forsciint

Laser printer attribution: Exploring new features and beyond

Anselmo Ferreira [a,*], Luiz C. Navarro [a], Giuliano Pinheiro [a], Jefersson A. dos Santos [b], Anderson Rocha [a]

[a] Institute of Computing, University of Campinas, Av. Albert Einstein, 1251, Cidade Universitaria, Campinas, SP CEP 13083-852, Brazil
[b] Universidade Federal de Minas Gerais, Department of Computer Science, Av. Antônio Carlos 6627 – Prédio do ICEx – Pampulha, Belo Horizonte, MG 31270-010, Brazil

ARTICLE INFO

Article history:
Received 26 August 2014
Received in revised form 26 November 2014
Accepted 27 November 2014
Available online 23 December 2014

Keywords:
Printer forensics
Texture patterns
Banding

ABSTRACT

With a huge amount of printed documents nowadays, identifying their source is useful for criminal investigations and also to authenticate digital copies of a document. In this paper, we propose novel techniques for laser printer attribution. Our solutions do not need very high resolution scanning of the investigated document and explore the multidirectional, multiscale and low-level gradient texture patterns yielded by printing devices. The main contributions of this work are: (1) the description of printed areas using multidirectional and multiscale co-occurring texture patterns; (2) description of texture on low-level gradient areas by a convolution texture gradient filter that emphasizes textures in specific transition areas and (3) the analysis of printer patterns in segments of interest, which we call frames, instead of whole documents or only printed letters. We show by experiments in a well documented dataset that the proposed methods outperform techniques described in the literature and present near-perfect classification accuracy being very promising for deployment in real-world forensic investigations.

1. Introduction

The massive use of printers is now giving rise to questions about authenticity of printed documents. Today, unknown contractual terms can be added easily and a forged correspondence can be linked to an innocent. Also, documents related to crimes such as child pornography photos, fake travel tickets, terrorist plots, fake money, pirated copies of books and illegal drug selling accounting are constantly printed everywhere. Identifying the source printer of these documents is an important clue to pinpoint their owner.

To understand the clues given by these printers and use them to identify the printer source, it is paramount to understand how they work. One of the most used printer devices currently is the Laser Printers (LPs). These devices work by using electromagnetic energy created by a laser canon onto fix the toner to a paper. As described by Chiang et al. [1], identifying the source of a printed document involves two strategies: the first, known as finding the *extrinsic signatures*, is an active procedure and involves embedding a signature on the printed page. This is done by modifying the document before it is sent to the printer or by encoding identification information, such as the device's serial number. The second, and most used way of identifying the source printer, is finding the *intrinsic signatures*. This is a passive strategy which is used on a scanned version of the document. It requires an understanding and modeling of the device mechanism to find clues in the printing pattern that are present on the scanned image. Most techniques applied to identification of laser printers take into account an artifact commonly caused by the printer manufacturing process: the *banding*. These techniques investigate how the texture in letters of text behaves and link it to a specific printer. Most of them [2–7] select a common letter in the text and describe the texture on it.

In this paper, we propose three solutions aimed at the identification of the source printer of a document that explore these intrinsic signatures. The proposed solutions do not need very high resolution digital versions of documents and take into account that this problem requires multidirectional and multiscale analysis, because of different printing patterns yielded by different manufacturing processes. The proposed solutions described in this paper are:

1. Two descriptors, based on multidirectional and multiscale properties of texture micro patterns. These descriptors are

* Corresponding author at: Av. Santa Isabel 1125, Barão Geraldo, Campinas, São Paulo CEP 13084643, Brazil. Tel.: +55 1933080321.
E-mail addresses: anselmoferreira@ic.unicamp.br, anselmo.ferreira@gmail.com (A. Ferreira).

http://dx.doi.org/10.1016/j.forsciint.2014.11.030
0379-0738/

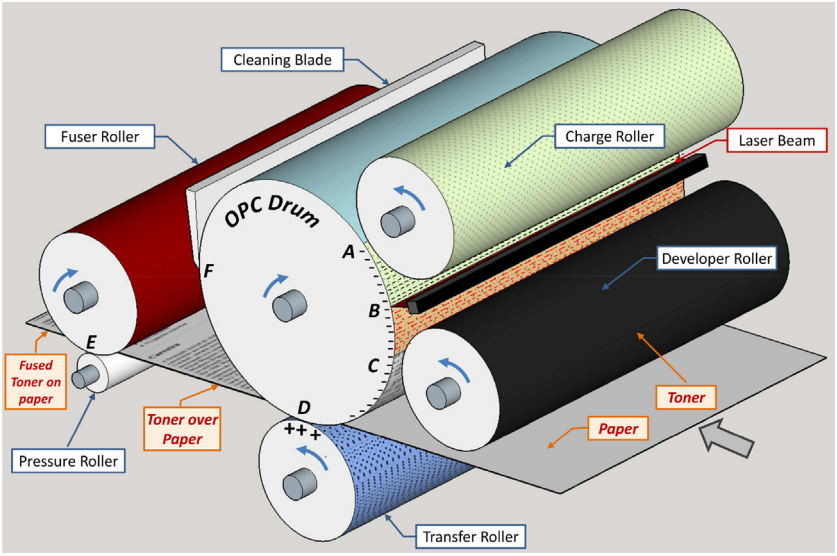

Fig. 1. Steps of LP workflow: (A) charging, (B) exposure, (C) development, (D) transfer, (E) fusing, (F) cleaning.

applied in text letters or regions of interest. These descriptors are focused on the inner part of printed letters.
2. Another descriptor, here described as the convolution texture gradient filter (CTGF). The CTGF is built as a histogram of low-level gradient filtered textures. We use filters of one or more scales, which are focused on filtering inner and outer parts of printed letters and figures.
3. The investigation of texture artifacts on segments of a document, called frames. With this approach, we can recognize the printing source of a document even if parts of it are unavailable or with problems. If the whole document is available, we can use this approach allied with fusion strategies, which provides even more reliable results.

We perform experiments in a well documented printed document benchmark, which is a very difficult one containing different letter sizes, styles and figures.

The dataset was created within the scope of this work and is freely available through FigShare [1] along with the source code of the proposed methods available on GitHub. [2]

Finally, we show that the presented techniques are very competitive and have important properties when compared to other ones in the literature.

2. How laser printers work

To understand the intrinsic signatures and how it can be detectable for laser printer attribution, the Laser Printer process must be known first. Laser printers basically use the attraction of opposite electrical charges in the printing process. The main component of the LP system is a revolving drum or cylinder. This assembly is made of photo-conductive material, which is discharged by light photons of a laser beam. As described by Chiang et al. [1], Laser Printers works in six steps:

1. **Charge:** the revolving drum that rotates at a constant angular velocity is positively charged by a roller or wire having electrical current moving through it.
2. **Exposition:** as the drum revolves, the printer uses a laser beam reflected by a mirror to discharge certain points on the drum, which will be the letters and images to be printed.
3. **Development:** after the pattern has been created on the drum, the printer coats these areas with positively charged ink (or toner) particles.
4. **Transferring:** the printing is done by moving the positive toner particles on the drum to a sheet of paper negatively charged, which moves on a belt below it.
5. **Fusion:** a fuser uses pressure and heat to fuse toner onto the paper.
6. **Cleaning:** to print the next page, a blade cleans the drum to eliminate any residual toner.

Fig. 1 depicts how LPs work.

In black and white printed documents, colors are represented by grayscale using standard conversion formulas to preserve visual perception characteristics, such as luminance. As laser printers have only one ink that is darkest black, grayscale intermediate tonalities are achieved using density variation from black and white small areas (above human eyes resolution) using halftones. Halftones are an old printing technique consisting in black small dots with different diameters over a white surface, which creates a grayscale visual illusion. Common halftone algorithms are error diffusion [8] and clustered dot halftone [9].

As laser printers are electromechanical devices with moving parts, there are many small physical differences on LPs such as motor drifting and gear precision that can be seen on printed pages. These informations patterns can be used as intrinsic signatures of these devices. The *banding* [10,11] is an artifact detectable on scanned printed images that can be used to identify the source printer. It is defined as nonuniform light and dark lines perpendicular to direction in which the paper moves through the printer.

Different printing devices have almost unique banding frequencies, depending of model and brand. To recognize this property, several techniques proposed in the printer attribution literature analyze the frequency domain of one dimensional signal of large halftone regions of the document. Studying the Fourier transform of the printed material can be useful to identify the frequencies at which printers work. But those features are only detected at higher resolutions, where variations on distances of halftones can be measured properly. In text documents, whereby only the black color is visible, the absence of halftone areas makes it difficult to perform the Fourier analysis of a signal. In this case, the banding can be seen as textures in specific characters and

[1] http://dx.doi.org/10.6084/m9.figshare.1263501
[2] https://github.com/anselmoferreira/printer_forensics_source_code

happens because of toner variations in the *development* stage of the LPs process. This variation is caused by electromechanical imperfections in LPs. We discuss in the next section techniques in the literature which aims at identifying the source printer of documents, using these intrinsic signatures discussed in this section or by extrinsic signatures, which can be understood as visible or invisible watermarks on the printed paper.

3. Existing solutions for laser printer attribution

Although our focus here is on discussing Computer Vision approaches for investigating intrinsic or extrinsic signatures for laser printer attribution, they are not the only way to identify the laser printer source of a document. Investigation methods of questioned documents also include physical, microscopical and chemical techniques [12]. Physical marks due to traction mechanisms, traces of toner spread on the paper and electrostatic drum defects create patterns, which can identify specific laser printer devices. On the other hand, chemical components of toner, analyzed by chemical methods such as spectroscopy [13,14] and x-ray [15] provide information about toner manufacturer and also can be used for comparison with seized evidence materials. Microscopy can also show some patterns on the toner fusion and letter borders.

Some of these methods are destructive, as they require the use of samples extracted from documents on destructive experiments. Another aspect of those methods is that they normally require special laboratory devices, equipment, and also experts to prepare, manipulate and analyze the samples. This does not happen with the same extent with Computer Vision-based techniques, which require only a scanned version of the document and little supervision.

Several computer vision techniques proposed for laser printer attribution in literature use similar approaches. Some of them are *halftone-based* [16,17] and are applied only in color documents, which often use images. Other techniques are *texture-based* and are applied on text documents [2,6,3–5,18], whereby halftones are not present. There are other techniques which aim at identifying the printer noise [19–21], among others. Although this section gives a guided tour on solutions available in the literature for forensic printer attribution, the reader may also want to refer to [1,10,22,23] to find other methodologies and review works.

Ali et al. [2] used the projection of text characters in one dimension as simple features to identify laser printers. The printed document is scanned and a one dimension projection (pixel values) of letters "I" is used as features in a Gaussian mixture model machine learning classifier. These feature vectors have, by nature, high dimensionality. Therefore, this solution is tied to a Principal Component Analysis piece, which is used to reduce the dimensionality of these feature vectors.

Lee et al. [19,20] proposed texture analysis of noise in the specific case of color documents containing images. To detect these noise patterns, printed documents are scanned and converted to CMYK color space, where the K band is discarded. This color space was used to minimize distortion and because it is the color space used by printers. The noise of the CMY image is isolated by subtracting the original image CMY and CMY filtered by the Wiener filter. A feature vector is calculated in this noise image through the computation of statistics in five gray level co-occurrence matrices proposed by Haralick [24] in this reference image. These feature vectors are then used to feed a machine learning classifier. Elkasrawi and Shafait [21] also used the noise pattern to identify the printer, but their feature vector is based on statistics of the noise in the row and column directions.

Choi et al. [25] used the forensic analysis of Wavelet transform statistics in the RGB image and in the image converted to CMYK to identify the source of color documents. Tsai et al. [26] used a similar strategy, but the analysis only considers the RGB color space. Mikkilineni et al. [3,4] proposed the use of texture descriptors based on statistics of gray level co-occurrence matrices to identify the source of text documents. In this technique, documents are scanned at 2400 dpi with eight bits by pixel and letters "e" are extracted in windows of approximately 180×160 pixels. After that, 22 statistics of gray-level co-occurrence matrices are extracted per character. Each feature vector is classified individually, using a 5-nearest neighbors classifier. The final classification uses the majority voting of each character classification. This work was extended upon in [5] by using Support Vector Machines and in [18], whereby the authors proposed a solution based on clustering and Euclidean distance to identify documents as resulting from an unknown printer.

Bulan et al. [16] used the correlation between geometric distortions caused by laser printers to identify them. This artifact is detected by subtracting the area that a printer should print and the area that it effectively prints. The technique extracts geometric signatures by estimating the positions of dots in halftone in printers on a training set and compared, by correlation, the positions of points in the test. Wu et al. [17] also used the geometric distortion to identify source printers. They modeled a projective transformation, which represents the geometric distortion, by using the center of letters in a scanned document and its image (TIFF) version. This model is solved by least squares using singular value decomposition and removal of outliers. A subset of the model parameters are then used as input feature vectors by a machine learning classifier.

Tsai et al. [6] combined the statistics of gray level co-occurrence matrices and sub-bands of wavelet transform. This was used in the particular case of identifying the laser printer source of a document which contains Chinese characters. The texture patterns extractions occur at a specific character of Chinese language after the documents are scanned. Jiang et al. [27] proposed the extraction of 9-d feature vectors from scanned documents based on Benford's law. These features are the first digit probability distribution of Discrete Cosine Transform coefficients from multi-size blocks.

Ryu et al. [28] proposed a solution to identify the halftone texture in color documents (or images). These documents are scanned in very high resolution (2400 dpi) and histograms of angles from Hough transforms are calculated in each CMYK band. These histograms are concatenated and yield a feature vector per document. These features are compared by correlation with reference patterns of a training set, composed by averaging the histograms of particular printers. The highest correlation will identify the source of the document.

Kee and Farid [7] proposed two solutions: the first is to find whether a document was printed by different printers and the second is focused on identifying the source of a document. To find forged documents (printed by more than one printer), they perform a well known graph-cut based clustering approach: the Normalized Cut [29]. They used letters of a document as the graph nodes that are clustered by the technique. To do what they call *ballistics* (identification of the source), they first choose a reference character. Then, similar letters are searched by template matching, preprocessed by histogram normalization and registered with the reference letter. The mean character $\overline{c}$ is calculated and Principal Component Analysis [30] is performed on the aligned characters. This yields the top p eigenvectors e_i, $i \in [1, p]$ which are used to calculate the printer profile, which is $P = \{\overline{c}, e_1, \ldots, e_p\}$. Given a test document, its letters and P of each printer are used to calculate a *reconstruction error*. The smallest mean error identifies the source of a printed document.

Gaubatz and Simske [31] proposed an authentication technique based on the presence of color tiles security deterrents on printed

text. Deterrents are watermarks (extrinsic signatures) on the printed paper and were previously proposed as an authentication methodology by Simske et al. [32]. This technique extracts statistical features from these deterrents and feeds a machine learning classifier. Mazzela and Marquis [33] studied text and dot-quality objective measurements to differentiate printed outputs.

Schreyer [34] used spatial and frequency analysis to identify source printers. This technique extracts statistical features in the noise image (mean, standard variation, correlation and mean squared error), in gradient image (mean and standard variation from the histogram), in the discrete cosine transformed image (mean and standard variation of coefficients at certain frequency sub-bands) and in the multi-resolution wavelet transformed image (mean and standard variation at different scales) and use them as feature vectors of machine learning classifiers.

Most of the literature methods presented thus far are limited in several ways. First, they are application-focused. In other words, they are applied on documents with text or documents with images. The second limitation is that they are applied only on text databases with the same font style and size. These particularities are not always useful when real-world documents, such as contractual clauses, are investigated. These documents usually have letters with different sizes, configurations (italic, bold, etc.), styles and also can contain figures. We believe that multidirectional and multiscale approaches are useful to laser printer source attribution in these cases. Other limitation of most of these techniques is the lack of a public benchmark for comparison. The techniques we introduce in the next section aim at solving such limitations.

4. Proposed methods for laser printer attribution

The techniques proposed in this paper were originated by a series of microscope analyses of printed documents. We investigated pictures (of same position on original document) of three letters from three documents, printed by different printers (they can be seen on Fig. 2). Although borders are more irregular and show more differences between printed characters, even on characters of the same printer, it is noticeable that inside the letters there are micro textures with different sizes and directions.

This investigation enforces our hypothesis that multidirectional and multiscale texture analyses are useful to identify the source printer. In documents with different font configurations, sizes, styles, and figures, the printing patterns are spread over different

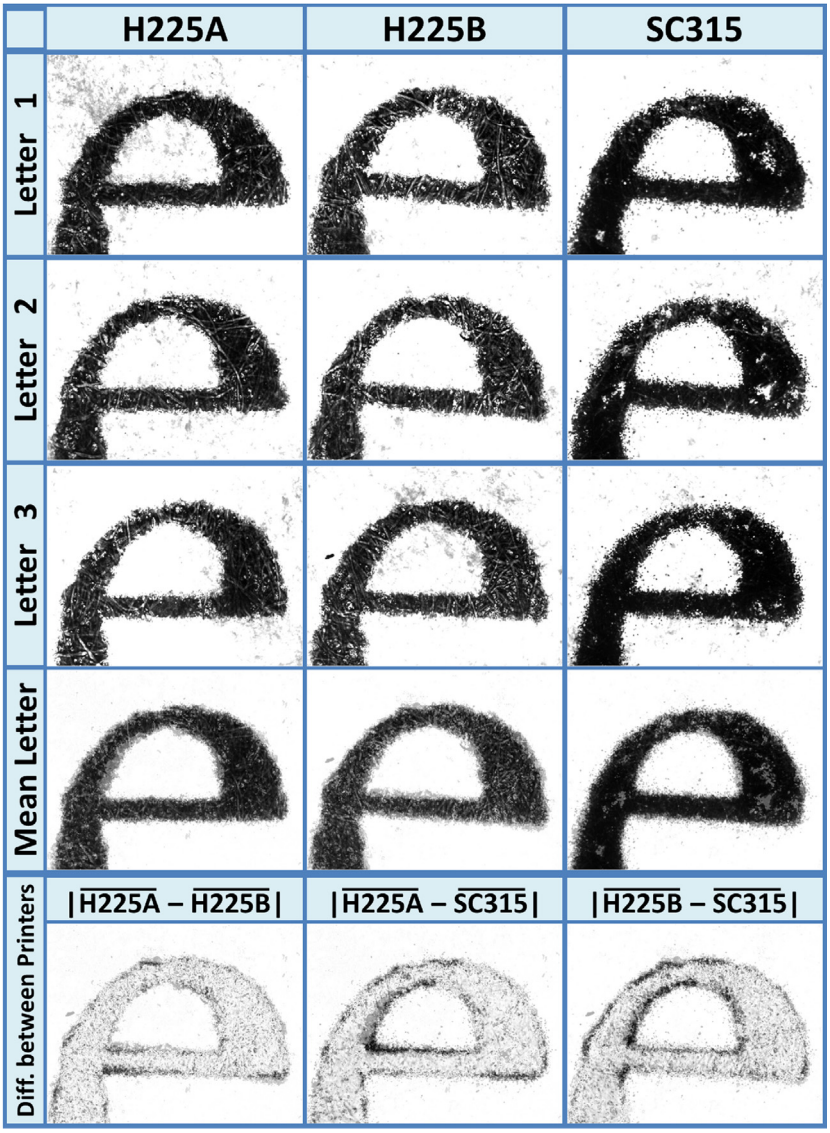

Fig. 2. Microscope images of three letters in three documents. The last row shows that the main differences are on the borders and some areas inside the letters (with low gradient). These are the regions which we aim at proper characterizing with the Convolution Gradient Texture Filter.

directions. Hence, the contributions of this paper for laser printer attribution are:

1. The analysis of multidirectional texture patterns captured through gray-level co-occurrence matrices, which is a set of statistics calculated over eight gray-level co-occurrence matrices, each one representing one texture direction
2. The multidirectional and multiscale approach, applied again over gray-level co-occurrence matrices. These two first approaches are applied inside the printing material (*e.g.*, letters), which is the area where the micro texture pattern is spread.
3. The convolution texture gradient filter (CTGF). This descriptor is created as histograms of filtered printing patterns over low-gradient areas. These areas are located commonly inside the printing material and close to the borders. We also extend this proposed approach to take advantage of multiscale filters, which increases the printing pattern investigated area. We use these low-gradient areas because they are intentionally created by printer firmwares to create visual effects not perceptible by the human eye. Investigating the pattern used by firmwares of different printers is useful to identify the source printer.

In the next subsections, we discuss the proposed methods in greater details.

4.1. Texture micro patterns via multidirectional gray-level co-occurrence matrices

Our first solution is based on statistics of gray-level co-occurrence matrices (GLCM), a well known micro-texture descriptor. Proposed by Haaralick et al. [24], these matrices are built by calculating how often two neighbor pixels i and j occur in a given direction and offset. Each direction will define a GLCM: West/East (0°), Southwest/Northeast (45°), South/North (90°) and Southeast/Northwest (135°). Fig. 3 depicts these directions.

After each of these four matrices are built, a set of statistics can be calculated to describe these textures. The original paper proposes 14 measures, such as the angular second moment, contrast, correlation, sum average and so on. For each measure, there are four values. The authors proposed to use the mean and range of these four values and, finally, a 28-d feature vector is used to describe the image.

Several GLCM variations have been proposed in the literature for printer attribution. We discuss the ones proposed by Mikkilineni et al. [3–5]. In these papers, the GLCM is calculated over a set of characters extracted from the printed document. The GLCM is calculated just in the pixels in a Region of Interest (ROI), which is the printed area of the rectangular region containing the letter. The authors use an offset (distance) of two pixels and build only one GLCM. The direction used in this case was only the pixels in the bottom side (270°). After that, 20 statistical features are calculated from the GLCM and two new metrics are proposed: the variance and entropy of pixel values in the ROI. At the end, 22 features are used in machine learning classifiers to identify the source printer. For more details, we refer the reader to A.

Differently from Mikkilineni et al.'s variation and the original GLCM, in this paper, we start by extending the basic GLCMs to incorporate eight angles (directions) in each pixel's neighborhood, using the original image scale. The eight GLCMs are built in the following neighboring directions: East (0°), Northeast (45°), North (90°) Northwest (135°), West (180°), Southwest (225°), South (270°) and Southeast (315°). After these matrices are built, we extract the same 22 statistical measures per GLCM proposed in Mikkilineni's et al. [3–5] approach. Hence, a $22 \times 8 = 176$-d feature vector is used to feed a machine learning classifier able to identify the source printer. Fig. 4 shows the neighboring directions used to build the proposed GLCMs.

4.2. Texture micro patterns via multiscale multidirectional gray-level co-occurrence matrices

Another GLCM variation proposed in this paper bears from the idea that multiple scales of a suspected document might spread uniquely the texture found in the original scale of each printed document. We propose a GLCM texture descriptor based on Gaussian Pyramidal Decomposition of images. The *Multiscale Multidirectional Gray Level Co-occurrence Matrices* are built in the same way as the Multidirectional GLCM presented before. The difference is the use of multiple scales of the original image in a Gaussian Pyramidal Decomposition. Using s scales, an $176 \times s$ feature vector is created.

This approach is, in part, inspired by Siqueira et al.'s work [35], where Multiscale GLCM descriptors are proposed. The core differences of our method and theirs are: (1) we do not use dimensionality reduction in feature vectors in order to preserve texture micro patterns found in different scales; (2) the data does not need to be normalized as in their work; (3) we use more directions (eight) in the GLCM construction in order to capture more subtleties of texture micro-patterns; (4) we consider $s = 4$ scales with a very particular configuration: two downscaled versions, one upscaled version and the original scale of the image. We chose this configuration because the low frequency components at these scales are more interesting for texture description after analyzing the microscope studies we carried out. Fig. 5 illustrates how the multidirectional and multiscale GLCMs source printer detector works. Afterwards, a classifier can be trained to

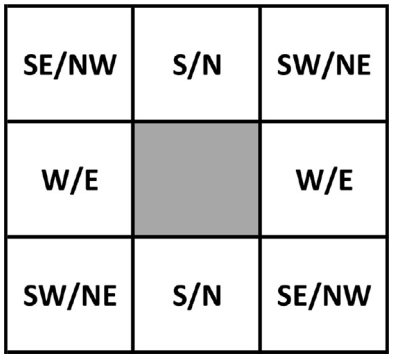

Fig. 3. Neighboring directions used to build the four gray level co-occurrence matrices proposed by Haaralick et al. [24]: West/East (0°), Southwest/Northeast (45°), South/North (90°) and Southeast/Northwest (135°).

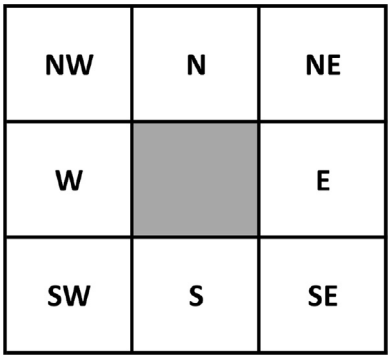

Fig. 4. Proposed multidirectional GLCM. We used statistics over eight matrices as printer texture signatures. Each matrix represents eight possible directions on each pixel's neighborhood: East (0°), Northeast (45°), North (90°) Northwest (135°), West (180°), Southwest (225°), South (270°) and Southeast (315°).

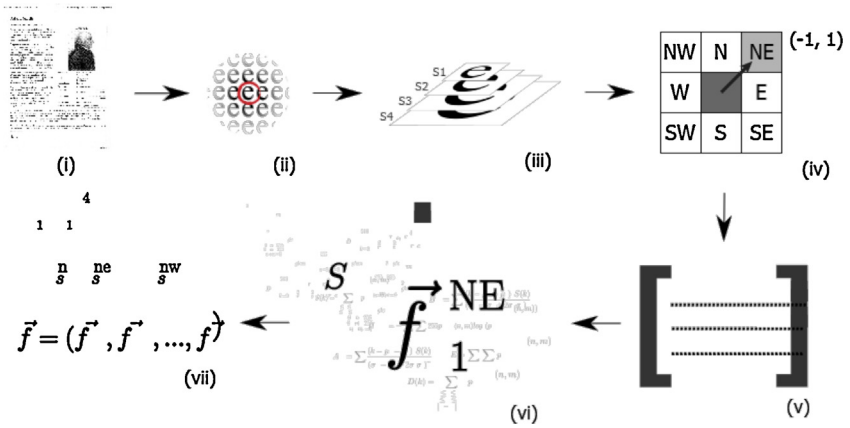

Fig. 5. Proposed multiscale and multidirectional GLCM. (i) scanned document; (ii) character extraction; (iii) Gaussian pyramidal decomposition; (iv) directions used in the multidirectional approach; (v) GLCMs construction in each direction at each scale; (vi) GLCMs statistical features extracted per scale and direction; (vii) final feature vector comprising all statistics extracted across different scales and directions.

identify a particular printer based on the texture of the printed material.

The GLCM approaches presented here are applied to inner area of printed text. These are the areas with multiple directions and scales micro texture behavior shown in Fig. 2. The Gaussian filter of a pyramidal decomposition will emphasize these inner areas by filtering just low frequency components at each scale. We believe that the analysis of these low level components yields a better printer attribution approach. Next section presents another proposed technique that is also applied in multiple directions and scales and works on areas with low gradient.

4.3. Texture micro patterns via convolution texture gradient filter

Textures on almost flat areas (with small gradient value) are intentionally generated by the printer firmware by combining near pixel values below human eye resolution. These textures are created to give tonalities impression, smoothing of borders, shadows, roughness, gradient or glossy effect. On the other hand, effects of mechanical parts can produce texture patterns near printer resolution, such as motor drift, gear backlash, laser focus, mirror imperfections, drum surface defects, among others.

Our third approach for laser printer attribution relies on the analysis of these low-level gradient areas. The proposed descriptor, the convolution texture gradient filter (CTGF), aims at describing the texture of low-gradient areas. Given a labeled training set of documents, CTGF learns a set of $n \times n$ pixel patterns (texture) in low-gradient areas that appear more frequently in a given printer, but not in others.

Given the scanned printed document S with size $r \times c$, the following transformations (summarized in Fig. 6) generate the feature vector of the proposed technique used for the learning and attribution process.

4.3.1. Negative

As a pre-processing step, the image pixels in S are inverted. Thus, values close to zero will mean white pixels and 255, black pixels. This is made for convenience in the algorithm operations and yields a negative image N.

4.3.2. Crop borders

In order to eliminate scanning noise at the image borders generated by external light, folding, among others, the negative image N is cropped, eliminating 6% of pixels in each border. This percentage in a letter paper document (216 mm × 279 mm), for example, corresponds to 12.96 mm × 16.74 mm margins, which covers areas with no printed information in typical documents. The negative cropped image is now denoted as matrix R. We still consider the dimensions of matrix R as $r \times c$ for convenience.

4.3.3. Micro texture patterns matrix

Textures with $n \times n$ neighbor pixels contained in R are then represented by two properties, which can be computed in parallel: their sum and maximum gradient between the central pixel and its neighbors. Although those two properties do not identify specific textures, they group textures of interest and allow filtering printer signatures. The convolution of R with an $n \times n$ matrix full of ones O results in the micro texture patterns matrix C

$$C = R * \mathrm{O} \tag{1}$$

where $*$ is the discrete convolution operator and

$$\mathrm{O} = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}_{n \times n} \tag{2}$$

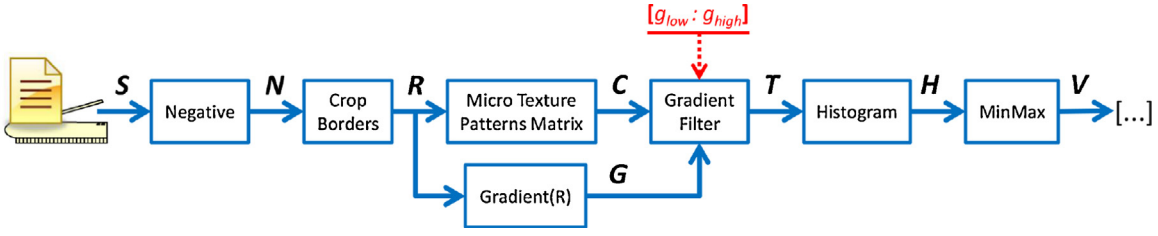

Fig. 6. Proposed solution for laser printer attribution using the convolution texture gradient filter.

Therefore,

$$C(i,j) = \begin{cases} 0 & \text{if } i = 1 \text{ or } i = r \text{ or } \quad j = 1 \quad \text{or} \quad j = c \\ R\left(i - \left(\frac{n-1}{2}\right) : i + \left(\frac{n-1}{2}\right), j - \left(\frac{n-1}{2}\right) : j + \left(\frac{n-1}{2}\right)\right) * \mathrm{O}, & \text{otherwise} \end{cases} \quad (3)$$

where $0 \le C(i,j) \le 255 \times n^2$.

4.3.4. Gradient (R)

In this step, we calculate the gradient of each pixel in R in a 3×3 area centered at the pixel to create the matrix of gradients G. The difference of two pixels x and y is calculated as

$$d_{x,y} = |x - y|. \quad (4)$$

Given the matrix R calculated previously, the gradient matrix G is calculated as

$$G(i,j) = \begin{cases} 0 & \text{if } i = 1 \text{ or } i = r \\ & \text{or } j = 1 \text{ or } j = c \\ \max\limits_{\substack{i-1 \le p \le i+1 \\ j-1 \le q \le j+1}} \left(d_{R(i,j),R(p,q)}\right) & \text{otherwise} \end{cases} \quad (5)$$

where $0 \le G(i,j) \le 255$.

4.3.5. Gradient filter

With gradients (G) and pixel sums (C), we filter the textures with gradients of interest. Two parameters (g_{low} and g_{high}) define the range of gradient range of textures that identify discriminant features for printer signature. Such parameters are selected from a training set of documents per suspected printer (which we will discuss in Section 5.6) for maximum results on the learning process. The matrix T of texture codes (sums) is then created by filtering textures that are not in the defined range. Those textures are the discriminant positions in the printed document. T is calculated according to Eq. (6).

$$T(i,j) = \begin{cases} C(i,j) & \text{if } g_{low} \le G(i,j) \le g_{\max} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

4.3.6. Histogram

Counting the number of positions for each texture in T from one (zero represents a position with no considered texture and is not used in the histogram) to $255 \times n^2$ generates the histogram vector with $255 \times n^2$ bins, as shown in Eq. (7).

$$\mathbf{H} = \mathrm{Histogram}(T, 1 : 255 \times n^2). \quad (7)$$

4.3.7. MinMax

The final feature vector $\mathbf{V}$, which represents the histogram of low-level gradient textures that a printer prints in the document is generated by applying a MinMax normalization on the histogram $\mathbf{H}$, scaling the components to the interval [0, 1], as Eq. (8) shows.

$$\begin{aligned} u &= Min_{\mathbf{H}(j)}(\mathbf{H}), \\ v &= Max_{\mathbf{H}(j)}(\mathbf{H}), \\ \mathbf{V}(j) &= \frac{\mathbf{H}(j) - u}{v - u}. \end{aligned} \quad (8)$$

As the final feature vectors are histograms of sums of pixels, they have $255 \times n^2$ dimensions, where n is the dimension of a squared sliding window used to calculate the texture.

This new method is based on $n \times n$ neighboring textures working with two basic properties: (1) sum of pixels; and (2) gradient filtering. The sum of pixels, obtained by a convolution with an $n \times n$ kernel of ones, measures the grayscale tone related to the visual impression of this region. The gradient is used to separate flat areas on text and images from the borders, as edge pixels have larger gradient than the interior of letters and background areas. Although those two properties cannot uniquely identify textures, they group textures of interest when used together, and also allow filtering printer signatures. Fig. 7 depicts how texture values vary for the same text and picture printed on different printers.

5. Experimental setup

In this section, we present the dataset and methodology used in the experiments. We discuss the experimental scenario, which parts of scanned documents are used in our investigation, metrics and how we implement the proposed methods and the state-of-the-art methods used for performance comparison.

5.1. Dataset

To validate the proposed techniques and compare them to the ones from the literature, we decided to use a dataset projected and built to provide instances of scanned documents as close as possible to a real situation. The databases used in prior works are limited in some way because they always consider fonts of same size and style, some of them have only text or only figures and some expect that the scanned documents are available in very high resolutions. Hence, the datasets in some prior works do not consider the case where the original document is not available to be scanned in very high resolution, just an already high-resolution digitized version of the document is available nor the computational cost of dealing with very high-resolution documents. This can possibly affect these approaches performance. In addition, for the works we surveyed, the used datasets are not readily available for download hardening comparisons using the same setups.

The aforementioned issues do not happen in the proposed dataset. We printed all or some of the 120 documents on ten LPs (showed on Table 1) in standard resolutions with Chamex white letter paper on 75 g/m^2 granularity, yielding 1184 TIFF images. These images are printable versions of Wikipedia documents converted to pdf with one, two or three pages and contain different letter sizes, fonts and figures. These documents were later scanned by a reference scanner (Plustek SO PL2546) at 600 dpi resolution and are separated by two factors: Language (English or Portuguese) and Figures (With or Without). The dataset is freely available on FigShare.[3]

[3] http://dx.doi.org/10.6084/m9.figshare.1263501

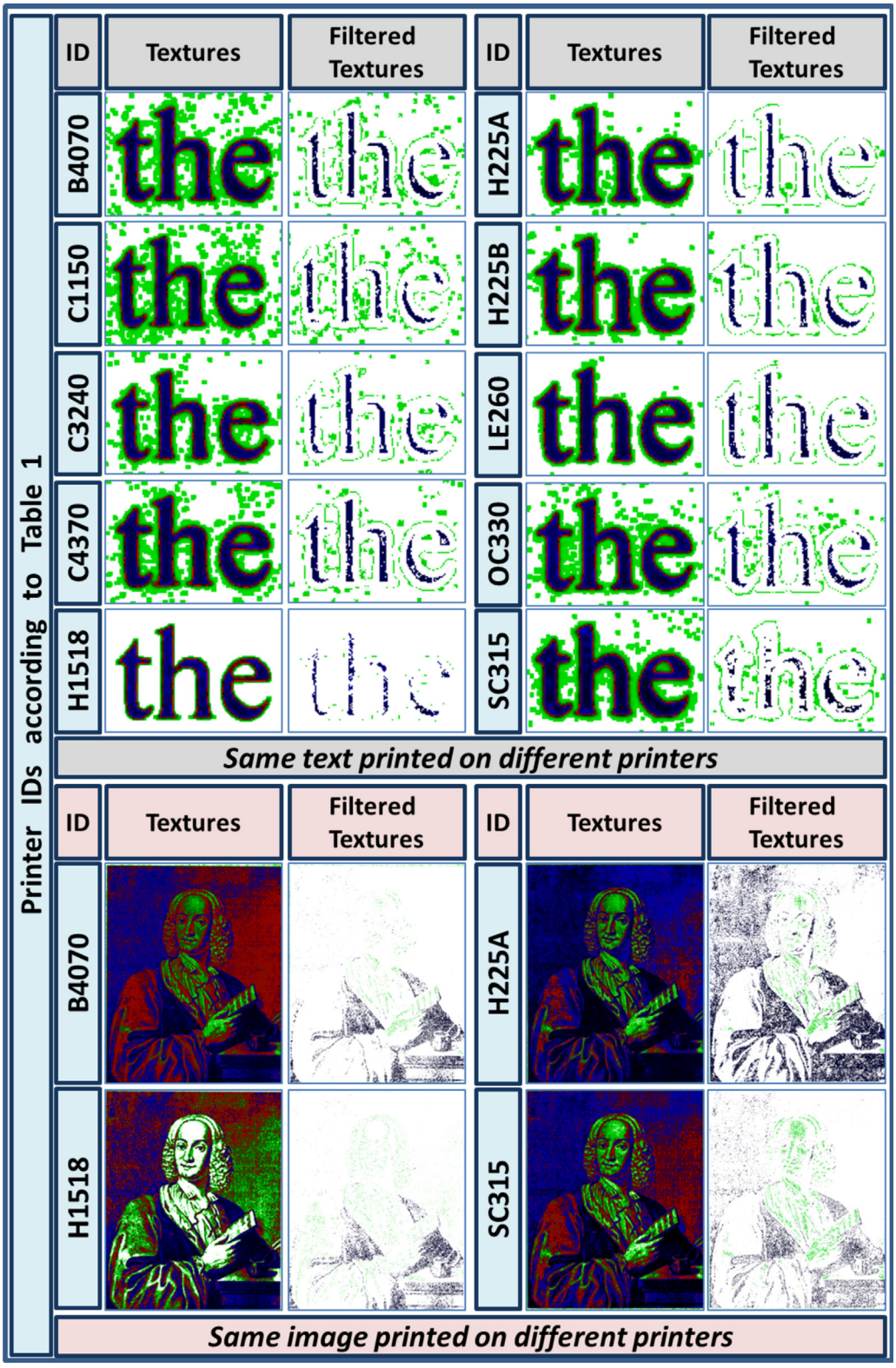

Fig. 7. Textures and filtered textures by gradient filter of (a) text and (b) image from different printers. White = 0, Green [1:765], Red = [766:1530] and Blue = [1531:2295]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2. Methodology

In this section, we discuss the background of the experimental methodology outlining the used regions of interest considered in each document, the metrics used, and implementation details about each method.

The techniques used in the experiments follow the pipeline presented in Fig. 8. Classifiers are trained with feature vectors yielded by different description techniques after the documents are printed and scanned at 600 dpi. Given one scanned printed document for testing, the classifier predicts its class. We have used the Support Vector Machines Classifier [36] with linear kernel in this process.

Table 1
Printers and number of documents per printer used in the experiments.

#	Printer ID	Manufacturer	Laser printer model	Number of printed documents
1	B4070	Brother	HL-4070CDW	120
2	C1150	Canon	D1150	116
3	C3240	Canon	MF3240	120
4	C4370	Canon	MF4370DN	120
5	H1518	Hewlett Packard	CP1518	120
6	H225A	Hewlett Packard	CP2025A	119
7	H225B	Hewlett Packard	CP2025B	110
8	LE260	Lexmark	E260DN	119
9	OC330	OKI Data	C330DN	120
10	SC315	Samsung	CLP315	120
			Total	**1184**

We used the one against one implementation of Support Vector Machines for multiclass problems. This approach works by building a set of $c(c-1)/2$ binary classifiers, where c is the number of available classes. Each of these classifiers will train data from each unique pair of classes. Then, at the end of the classification step, a voting strategy is performed. Each result of each binary classifier is considered a vote and the class with the maximum number of votes will be the classification of the given sample.

5.3. Sampling approaches

To study the effects of document sampling on the printer attribution process, we will discuss here the analysis on different areas of documents. It includes characters, frames and the whole document.

5.3.1. Characters

As texture analysis implies in the investigation of printed areas and their interactions with paper borders, we start by extracting specific letters from the digital versions of documents. These letter images will be the useful version of the dataset used in the experiments. To extract all letters from a document, we first implemented an algorithm that searches for connected components in graphs. Using thresholding and considering the neighboring pixels as graph nodes connected by neighbors, we extract useful masks on the image to capture how the letters look like.

To distinguish and separate the characters with higher occurrence (with same size and same font), we used a descriptor similar to Local Binary Patterns [37]. This descriptor separates a black and white version of the given image on slices of a superposed imaginary circle, describing them by counting the

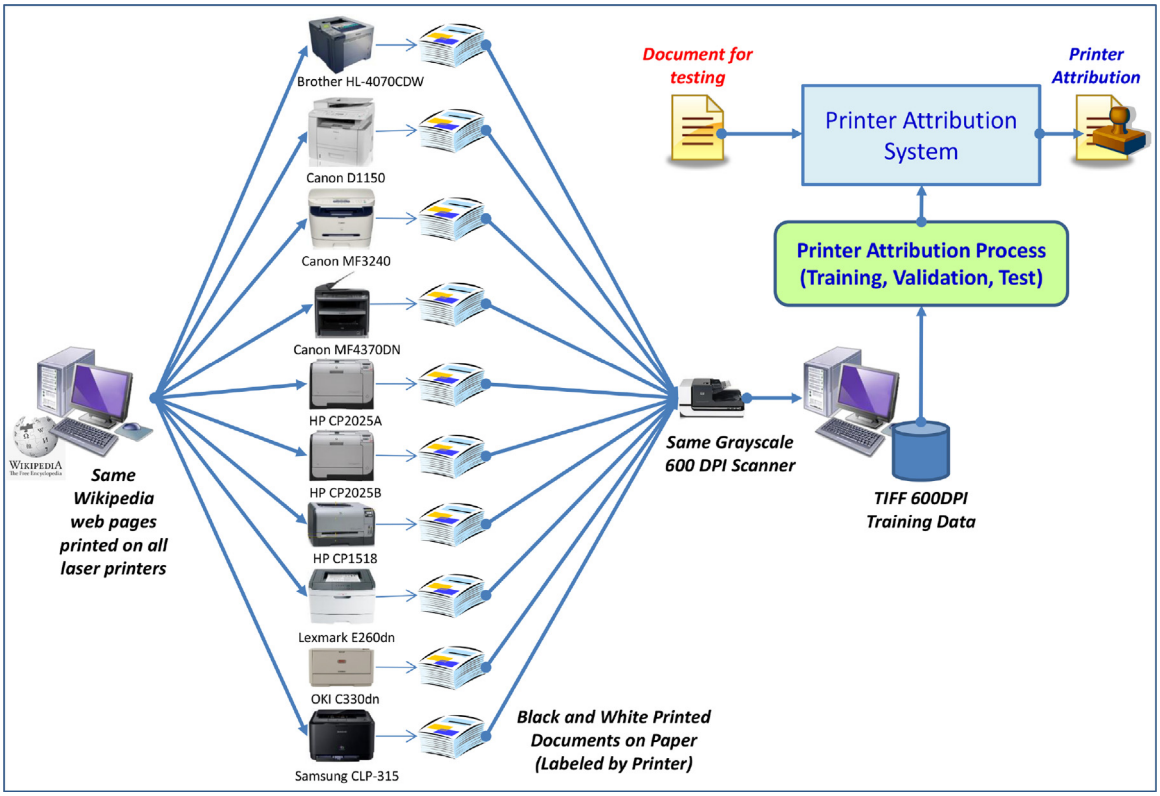

Fig. 8. Workflow used in the experiments. Firstly, the documents are printed by different printers and scanned. After that, a classifier is trained on feature vectors created through the different description techniques. Given an investigated document, the classifier will predict its class based on the trained models.

number of white and black pixels on each slice. The final descriptor is the counting of black and white pixels or ratios as black and white pixels density among others. This algorithm, when fed with a reference character as input, can separate with high hit rates the letters from the rest of extracted connected components. For this part, we focused on letters "e" as it is the most common letter in English documents and were also used before in the literature [7,3–5]. The final letter dataset has 245,000 extracted characters.

To classify the source of a given investigated printed document using this approach, the letters "e" are firstly extracted. Then, each letter is classified by a printer attribution method (e.g., the ones discussed in this paper), and a majority voting is applied in the end. The occurrence of each labeled class is counted on these letters and the most voted class will define the class of a document.

5.3.2. Frames

A letter paper, scanned in grayscale at 600 dpi without compression, produces a very large file with approximately 31 Mb of size, corresponding to about 5 K by 6.6 K pixels, even after discarding 6% on each border of the document corresponding to blank margins carrying external light scanning noise. After cropping, the remaining number of pixels is still very large (about 4.4 K by 5.8 K pixels). There are also areas inside the document that are completely blank (without printed ink). As those areas do not contribute with information about the printer, it is useful to split the large document in smaller samples, which maintain printer characteristics and can generate more feature vectors for the training and testing learning process.

In previous works [7,3–5], character samplings were proposed to capture texture behavior on printed documents. Letters "e", which is the most used letter in English texts, are extracted in each document. A mask of a template letter "e" is used to scan, compare and cut its copies from the document, capturing its pixels. The typical letter "e" in documents are inside an area of 40 × 50 pixels. This process is normally time consuming and not very accurate.

In this paper, we propose to use chunks of letters in regions of interest from a document, which we call *frames*. Frames are rectangular areas inside the document that have sufficient printed material to keep the characteristics of a printed document. The process used to obtain frames from the cropped images with 4.4 K by 5.8 K pixels consists of dividing them in five columns by six rows of frames, resulting in about 900 by 980 pixels corresponding to 37 mm (1.5″) by 43 mm (1.6″).

In order to avoid frames that do not contain enough printed areas, we state that the minimum accepted ratio between dark pixels (black and dark gray) and blank ones (blank and light gray) should be 0.02. This process eliminates frames that are completely blank or have only a few printed symbols on it. For reference, Fig. 9 shows the same document sampled by frame and character.

5.3.3. Document

To consider the scenario with just one frame and to evaluate the methods when not using any kind of voting scheme, we have also proposed an approach based on the whole document. Although there are several ways of describing a document using only one feature vector (e.g., each character GLCM can be accumulated to yield a single GLCM, from which one single feature vector can be extracted), we decided to apply the texture descriptors on the whole document, similar to some state-of-the-art techniques [21,27,34].

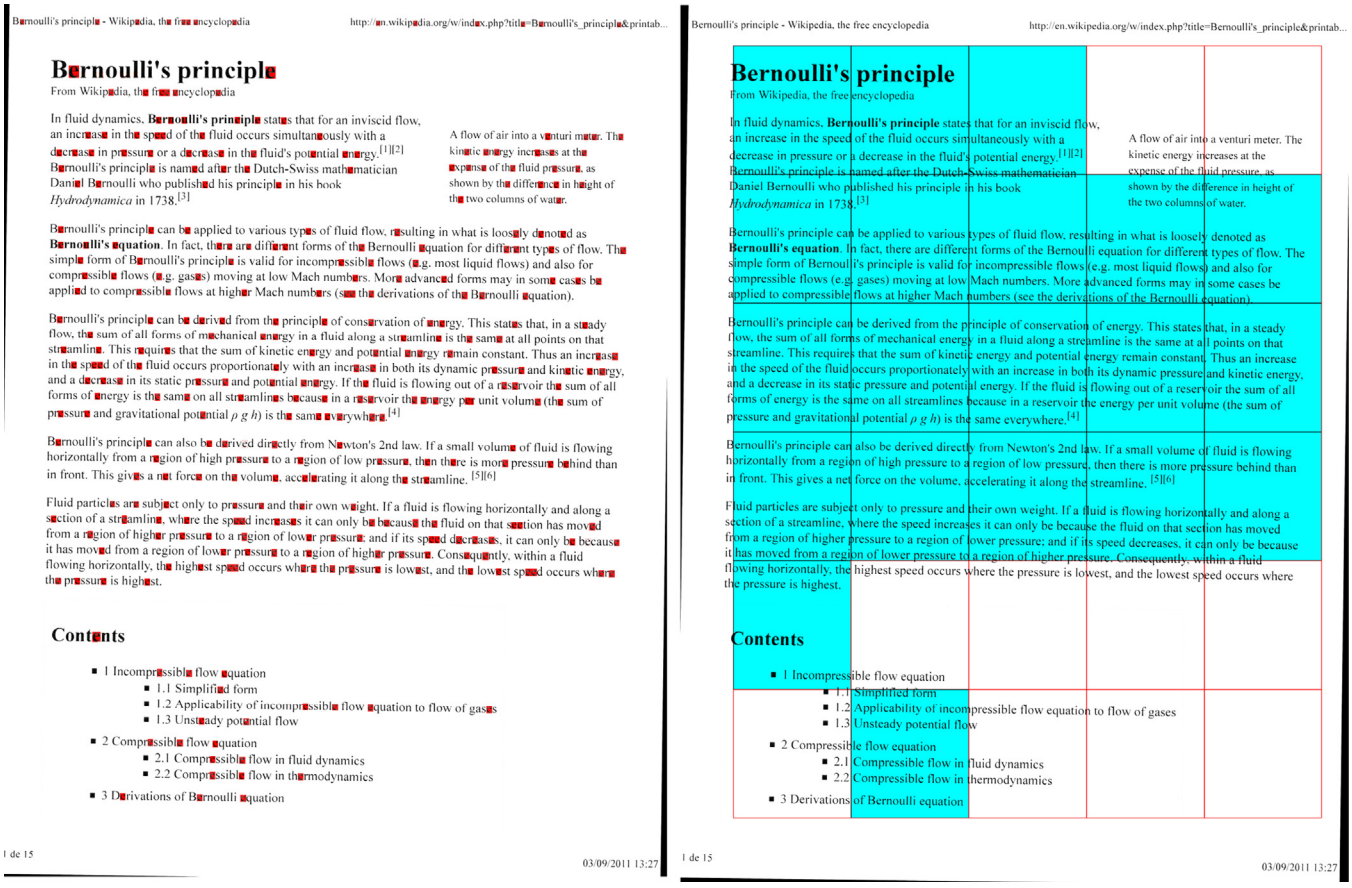

Fig. 9. Letter (left) and frame (right) sampling from a scanned document. The red areas identify the extracted letters while the cyan areas identify the extracted frames. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.4. Metrics and statistics

We adopt 5 × 2 cross validation protocol. Using this approach, five replications of the two-fold cross-validation protocol are performed. In each one, a set $\boldsymbol{X}$ is divided into $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ and a classifier is trained with $\boldsymbol{X}_1$ and tested on $\boldsymbol{X}_2$. Thereafter, training/testing sets are switched and the process repeated. There are 5 × 2 =10 different executions in the end. This is considered an optimal benchmarking protocol for learning algorithms [38].

We use a set of known metrics to assess all the algorithms performance using the above cross validation approach. In a multi-class problem with c classes, a confusion matrix M is built with c rows and c columns on each round of the 5 × 2 cross validation. Main diagonal values of M will show the right hits for each class. Other values are false hits

$$accuracy = \frac{\sum_{i=1}^{c} M(i,i)}{\sum_{i=1}^{c}\sum_{j=1}^{c} M(i,j)} \tag{9}$$

The precision of a given class (in this case, a printer) i, is defined as the fraction of events where the classifier *correctly* classified i out of all instances classified as being from that class

$$\boldsymbol{Precision}(i) = \frac{M(i,i)}{\sum_{j=1}^{C} M(j,i)} \tag{10}$$

The recall of a given class i is the fraction of events where the classifier correctly classified i out of all instances of that class

$$\boldsymbol{Recall}(i) = \frac{M(i,i)}{\sum_{j=1}^{C} M(i,j)} \tag{11}$$

The *f-measure* of a given class i considers both the precision and recall in the analysis. It can be interpreted as the harmonic mean of precision and recall, where it reaches its best value at 1 and worst score at 0

$$\boldsymbol{f}(i) = 2 \cdot \frac{\boldsymbol{Precision}(i) \cdot \boldsymbol{Recall}(i)}{\boldsymbol{Precision}(i) + \boldsymbol{Recall}(i)} \tag{12}$$

We perform a series of statistical tests to define if the results are statistically significant. First, we confirm if all techniques are statistically different (also known as pre-test). If they are, we check the techniques pairwise to define which ones are statistically different when compared to other (also known as post-test). Each of these steps usually involves a statistical test and a confidence level for the test. Here we consider a confidence level of 95% for each test. As pre-test, we consider the Friedmann test. This test is non-parametric and is used to determine if subjects change significantly across occasions and conditions. To compare the techniques pairwise (also known as multi-compare approach), we use the Tukey–Kramer approach (also known as Honestly Significant Difference (HSD).

5.5. Baselines

We compare our proposed techniques against four state-of-the-art methods (presented in Section 3) and also against two well-known texture descriptors widely used in content-based image retrieval applications.

The first state-of-the-art technique uses gray-level-co-occurrence matrices (which we call GLCM) applied to laser printer attribution, proposed by Mikkilineni et al. [3,4]. This technique describes the neighborhood behavior of pixels in a two-dimensional histogram given an offset, yielding one GLCM in which 22 statistics are calculated. The original GLCM of Mikkilineni et al. [3,4] uses an offset of $dr = 2$ and $dc = 0$ (dr stands for the offset in the rows while dc stands for the offset in the columns). In our implementation, we used dr in the interval $1 \leq dr \leq 3$ and we found the best as $dr = 1$. This is explainable because the Regions of Interest in our database are smaller than the ones in [3,4] approach. Although this technique was originally proposed to operate on characters, we also evaluate its performance on documents and frames directly. The 22 statistics extracted from the GLCM are discussed individually on Appendix A and are also used in our proposed GLCM variations.

The second considered method is based on statistics of Discrete Wavelet Transform (which we call DWT_STATS) from color bands applied to laser printer attribution, proposed by Choi et al. [25]. In this implementation, 39 statistical features are extracted from the HH Discrete Wavelet Transform sub-band per image. This approach is also applied document-, character-, and frame-wise.

The third method evaluated was the statistics of printer noise (which we call NOISE_STATS) in the row and column direction by Elkasrawi and Shafait [21]. This technique, based on a previous work of Khanna et al. [39] on scanners, works by first filtering the printed area with Otsu's threshold [40]. By binarizing the image with this threshold, the authors compute the median gray-level for the foreground as well as the median gray-level for the background pixels. Hence, a clean image is generated by only having gray-level values of all foreground and background pixels set to the median foreground and background calculated. The noise image is then obtained by subtracting the original image from the clean image. The mean of rows and columns of this noise reference image is calculated, yielding two vectors: the correlation between each row of the noise image and the average of all columns, as well as the correlation between each column and the average of all rows. Finally, a set of 15 statistics are calculated over these vectors. This approach is evaluated document-, character-, and frame-wise.

The last state-of-the-art method implemented was the technique proposed by Kee and Farid [7] (which we call RECONST_ ERROR). This technique has three steps: pre-processing, printer profiling and source identification. In the pre-processing step, the authors first choose a reference character (they chose the letter 'e'). Then, same letters are searched by template matching, preprocessed by histogram normalization and registered with the reference letter. In the printer profiling step, the mean character $\bar{c}$ per printer is calculated and Principal Component Analysis (PCA) [30] is performed on these aligned characters per printer. The printer profile are the PCA top p eigenvectors e_i, $i \in [1, p]$ and the mean character. In the source identification step, a test document is given, its letters 'e' are extracted and preprocessed the same way. These letters are used with the top p eigenvectors and mean character per printer to calculate a *reconstruction error* of each printer. The smallest mean error will identify the source of a printed document. This is the only method in the literature that does not use a known machine learning classifier. Therefore, we consider, in the 5 × 2 cross validation, the printer profiling phase as the training and the source identification as the testing step. This approach is only applied on characters.

In addition to the state-of-the-art methods considered herein, we also assess two well-known texture descriptors widely used in the literature. The first one is the Local Binary Patterns (LBP) [37] and the Histogram of Gradients [41] (HOG). The LBP is a histogram of eight-neighboring pixel relations. HOG consists in histograms of gradient orientations in localized regions (rectangular or circular) of an image. We chose these descriptors because they can be regarded as multidirectional descriptors.

5.6. Implementation aspects of the proposed methods

We first consider the two proposed GLCM variations: the multidirectional and the multidirectional and multiscale ones. We also implement four CTGF variations, three exploring 3×3, 5×5, and 7×7 filter sizes and a multiscale one exploring all the previous filter sizes.

The multidirectional GLCM hereinafter referred to as GLCM_MD consists of 22 statistics calculated on each GLCM built using one neighboring pixel offset over eight directions, as described in Section 4.1. The final feature vector has $22 \times 8 = 176$ dimensions. The multidirectional/multiscale (GLCM_MDMS), in turn, consider four scales of the image Gaussian pyramidal decomposition: the original scale, two down-scales and one up-scale. The final feature vector lies in the $176 \times 4 = 704 - d$ space.

CTGF is built as described on Section 4.3 and yields feature vectors with $3^2 \times 255 = 2295$ ($n = 3$), $5^2 \times 255 = 6375$ ($n = 5$) and $7^2 \times 255 = 12,495$ ($n = 7$) dimensions. We also evaluate a combined approach, in which we consider the different scales in a combined form creating what we call the *Multiscale CTGF* (hereinafter referred to as CTGF_MDMS), with $2295 + 6375 + 12,495 = 21,165$ dimensions. These feature vectors undergo dimensionality reduction on each filter window size as we shall discuss later in this paper.

Finally, we test the complementarity of the proposed methods by fusing the feature vectors from the CTGF using the 3×3 mask and GLCM_MDMS, creating what we call the CTGF_GLCM_MDMS.

6. Results and discussion

We now turn our attention to the actual experiments and results. We start with a study on dimensionality reduction for the CTGF method as one could wonder if all its features are really necessary for attribution. Then, we present the experiments for all methods considered herein followed by a proper statistical analysis of the results.

6.1. Convolution texture gradient filter parameters and dimensionality reduction

The main parameters of CTGF method are (g_{low}, g_{high}), which are defined during training. For that we consider the 5×2 cross validation protocol discussed in Section 5.4.

We performed two experiment configurations: keeping $g_{low}=1$ and varying g_{high} and keeping $g_{high}=254$ and varying g_{low}. The experiments were performed frame-wise. For this experiment, we used the one-against one multiclass SVM with linear kernel and the CTGF filter window size was set to 3×3. Fig. 10 shows the results.

Fig. 10 shows that keeping g_{low} as 1 and reducing g_{high} from 128 to 16 (solid blue line) produced very close results on the 5×2 cross-folding validations. In addition, in such situations, the classification differences are not statistically significant according to Friedman statistical tests. When tests are performed keeping g_{high} at 254 and varying g_{low} from 1 to 128, then best results are in the interval which included gradient values over the interval (1, 32).

The experiment discussed in this paper shows that important texture information for printer attribution is inside the gradient interval (1:32) for CTGF. This is explainable by grayscale jitters (*i.e.*, grayscale noise) on flat black and white areas of printed document due to printing variations (positioning, backslash, toner development, *etc.*). It is also important to understand that variations of gradient in the range (1, 32) on grayscale neighbor pixels are practically undetectable at normal resolution (600 dpi) for the human eye. Filtering texture values by a convolution window in this gradient interval around flat color areas creates a highly discriminative noise signature.

The result of (g_{low}, g_{high}) filtering by the proposed technique may result in some components that are not significant for the attribution process and a dimensionality reduction approach can be applied. We use a simple dimensionality reduction method that discards dimensions where the distance between its maximum and minimum values (also known as range) is less than the mean of

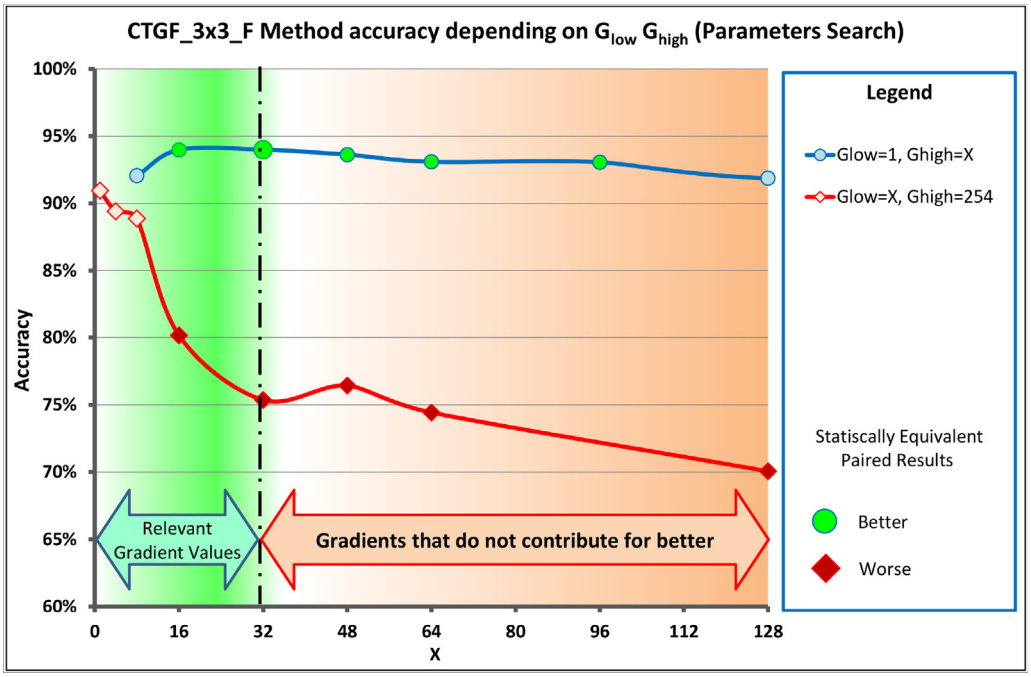

Fig. 10. Filter parameter search for the proposed convolution texture gradient filter. (For interpretation of the references to color in text, the reader is referred to the web version of this article.)

the overall components distance, calculated during training. This yields a binary vector (***KeepVector***), which is used to eliminate or keep features from feature vectors used for classification.

To find the vector ***KeepVector***, we describe a training set of documents with CTFG and put all the feature vectors ***V*** in a matrix F. Then, we calculate the mean of components range (component or dimension here can be seen as each column of matrix F). The range of a component is defined as the subtraction between the maximum and minimum value of that component (or column of matrix F). After this, we calculate the mean and build a binary vector ***KeepVector*** with $255 \times n^2$ dimensions. This vector is used in the feature vector construction, indicating whether a dimension in a new feature vector will be kept (its range is higher or equal the mean of dimensions range of matrix F) or not. ***KeepVector*** is built as Eq. (13) shows. Fig. 11 shows an example of the feature vector dimension reduction process and Fig. 12 shows the mean reduced feature vectors of some printers used in this work.

$$\boldsymbol{KeepVector}(i) = \begin{cases} 1 & \text{if } Max_{\boldsymbol{V}(i)}(i) - Min_{\boldsymbol{V}(i)}(i) \geq R \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

where

$$R = \frac{\sum_{i=1}^{255\times n^2} Max_{F(:,i)}(i) - Min_{F(:,i)}(i)}{255 \times n^2} \quad (14)$$

To validate the proposed dimensionality reduction technique, we also conducted experiments using 5×2 cross-validation experiments comparing it to PCA. The PCA results are not as good as the ones obtained with the aforementioned method.

The principal components of the CTGF histogram are more related to the structure of the image pixels than the intrinsic noise used to differ printers. Therefore, PCA ends up discarding components otherwise useful for printer attribution, that is why it does not perform so well in comparison with the feature selection method discussed above. Table 2 shows the comparison of the dimensionality reduction methods tested.

As Table 2 shows, the proposed dimensionality reduction technique selected the most important dimensions of feature vectors for classification, achieving a mean accuracy of 94.44%, against the best PCA best configuration, which achieved a 92.82% mean accuracy considering the cross-validation procedure adopted. The results are explainable as the proposed method eliminates dimensions with small variation in the training stage, not performing any additional linear transformation on the data. This dimensionality reduction approach, when applied in CTGF on frames, keeps 638 dimensions of feature vectors for the CTGF with a 3×3 mask, 2660 dimensions of feature vectors for CTGF with 5×5 mask, 6672 dimensions of feature vectors for the CTGF with 7×7 mask and 638+2660+6672=9970 dimensions for the multidirectional and multiscale CTGF. When applied on

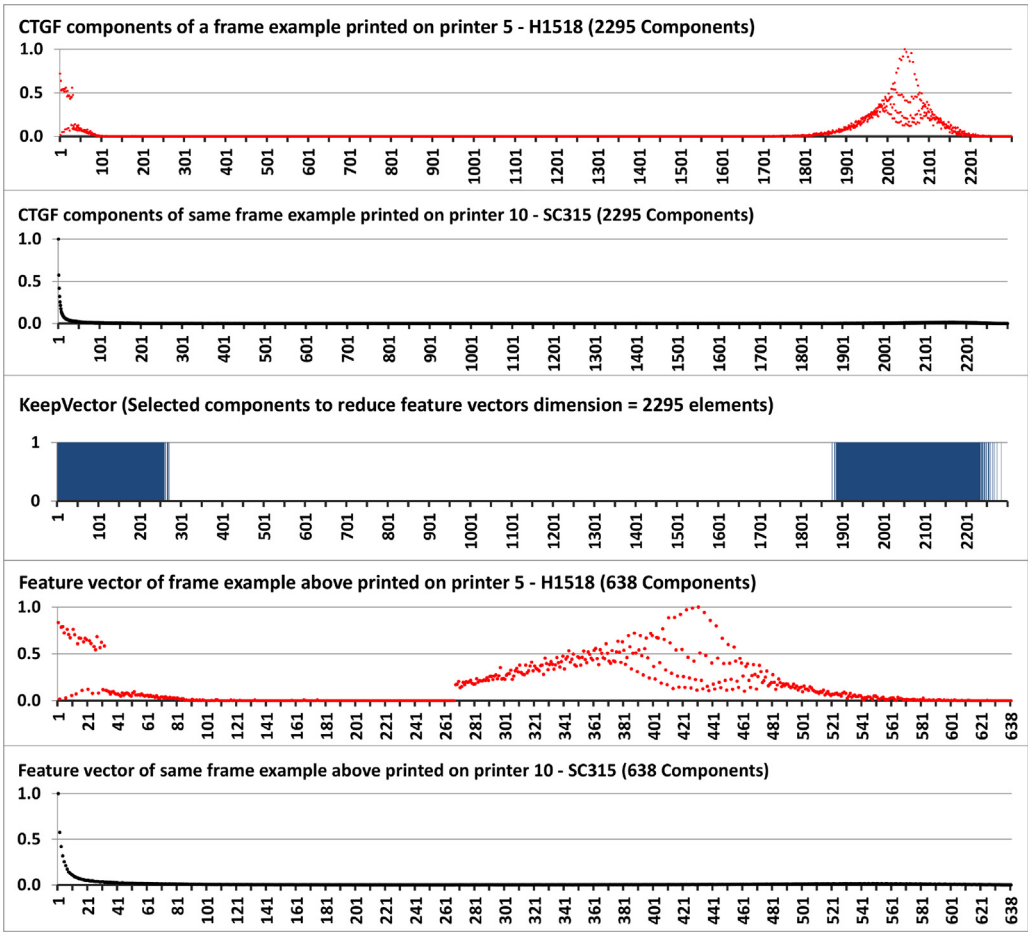

Fig. 11. Feature vector reduction process. The first and second rows show examples of feature vectors calculated using the CTGF approach with a 3×3 filter size. These feature vectors were calculated on the same document subset (which we call frames) printed by two different printers. The third row shows the binary vector ***KeepVector***, in which the colored regions indicate what dimensions from the 2295 must be kept. The fourth row shows that the reduced feature vectors must have 638 dimensions. The fifth and sixth rows show the reduced feature vectors from the two printers showed on first and second rows. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

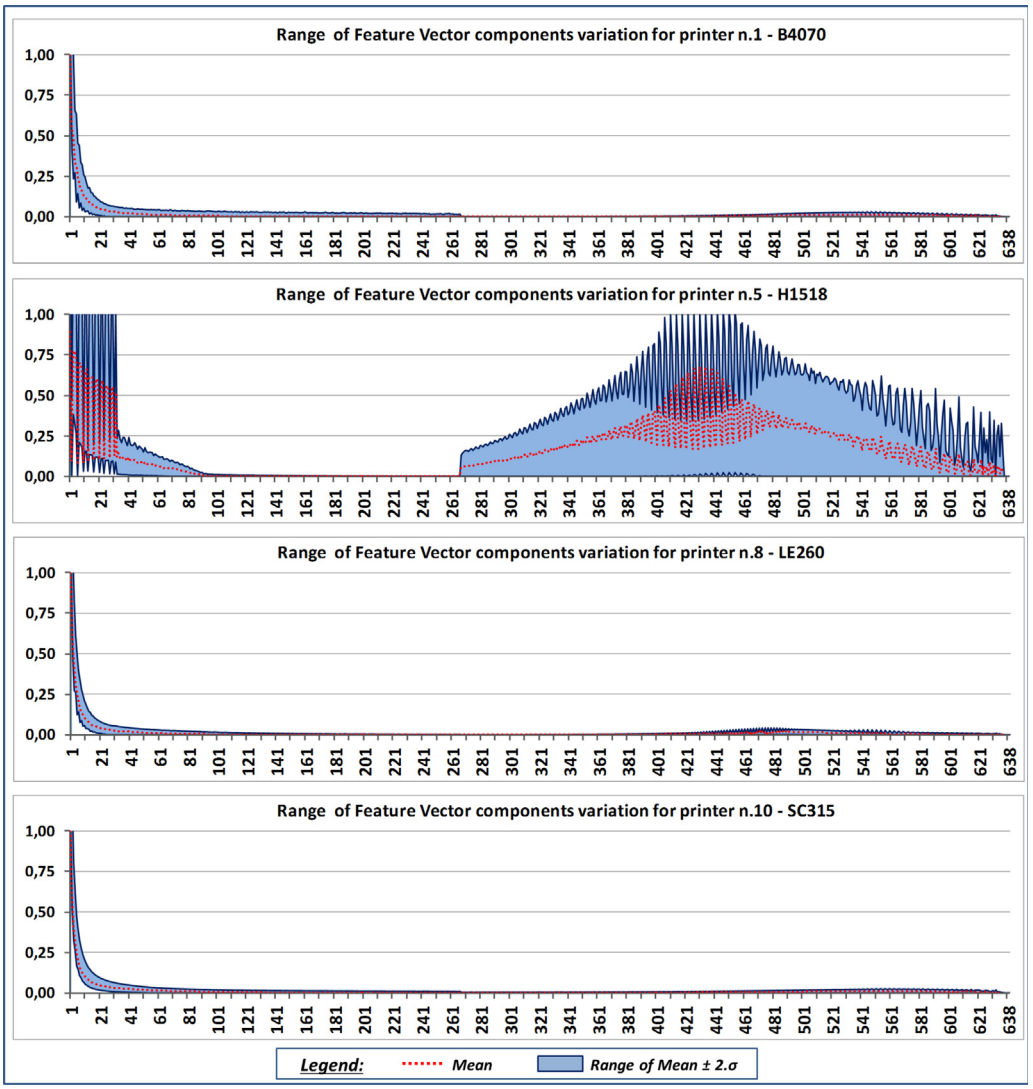

Fig. 12. Printer signatures of some printers using the proposed CTGF with a 3×3 filter size and the proposed dimensionality reduction approach.

documents, it keeps 471 dimensions of feature vectors for the CTGF with 3×3 mask, 2347 dimensions of feature vectors for the CTGF with 5×5 mask, 4842 dimensions for the 7×7 mask and 471+2347+4842=7660 dimensions for the multidirectional and multiscale CTGF.

6.2. Laser printer attribution experiments

With the dataset presented in Section 5.1 and methodology described in Section 5.2 in mind, we now discuss the experimental results, whereby we validate the proposed approaches against the

Table 2
Comparison of the proposed dimensionality reduction approach with some PCA variations. We used $n = 3$ for the CTGF filter size, in which the standard feature vector has 2295 dimensions.

Method CTGF_3 × 3_F dimensionality reduction test results	Feature vector site (original 2295)	Accuracy (%)			
		Min	Mean	Max	Std. Dev.
Component (max–min) ≥ Mean (max–min)[a]	**638**	93.17	**94.44**	96.59	1.03
PCA Sum of Eigenvalues = 99.9%[b]	**311**	90.29	**92.82**	95.06	1.48
PCA Sum of Eigenvalues = 100%	**2281**[c]	89.08	**91.53**	93.19	1.38
PCA sum of Eigenvalues = 99%	40	88.06	90.49	93.17	1.80
PCA sum of Eigenvalues = 95%	2	33.56	35.04	36.69	0.95
PCA sum of Eigenvalues = 90%	1	20.95	23.43	26.07	1.54

[a] Means that each selected component has max–min distance ≥ Mean of overall component max–min distances.
[b] Sum of Eigenvalues = x means that eigenvalues sum of selected components does not exceed x.
[c] Components wich eigenvalues = 0 are discarded.

state-of-the-art methods. Table 3 shows experimental results considering the 5 × 2 cross-validation protocol. We applied the techniques on characters, documents and frames as described in Section 5.3.

As expected, the worse experiment was DWT_STATS [25]. This happens because this technique was proposed for color documents, operating on RGB and CMYK color bands. NOISE_STATS [21] showed its best accuracy (68.86%) for characters. In this case, the printer noise is extracted from the letters in a region with small background perturbation. For frames and documents, this technique showed worse results (42.26 and 38.81%), as a large background area is considered hardening the noise estimation.

The RECONST_ERROR [7] was proposed to work only in characters, then we applied this proposed technique only in the extracted text letters and it yielded a classification accuracy of 84.86%.

GLCM [3,4] was the state-of-the-art method which yielded the best results. Although it was originally proposed to operate on characters, on frames and documents it also showed decent classification accuracies (93.62 and 82.56%, respectively). On characters, it yields the best classification accuracy for a method proposed in the literature: 94.19%.

The LBP [37] and HOG [41] approaches are general-purpose texture descriptors but they also showed decent classification results. HOG yielded a 95.79% accuracy for characters, 74.35% for frames and 79.66% for documents. LBP yielded 90.20% classification accuracy for characters, 95.20% for frames and 88.07% accuracy for documents. These good results have a reason: HOG uses a histogram of gradients, hence it identifies the printing process artifacts between the text and background (borders). LBP uses a histogram of relations between a pixel and its neighbors that also enables the identification of printer patterns in a muldirectional way.

Table 3
Mean Accuracies of 5 × 2 cross validation applying the proposed and state-of-the-art techniques on characters (C), frames (F) and documents (D). The proposed techniques in this paper are the ones in bold in the column "Methods".

Method	Accuracy Statistics on Crossfolding 5x2 Experiments					
	Mean	σ	Mean-2σ	Mean+2σ	Min	Max
CTGF_GLCM_MDMS_F	**98,47**	**0,60**	**97,26**	**99,67**	**96,93**	**99,15**
GLCM_MDMS_F	**98,38**	**0,72**	**96,95**	**99,81**	**97,10**	**99,32**
GLCM_MD_F	97,15	0,84	95,47	98,84	95,40	98,30
GLCM_MDMS_C	**97,60**	0,72	**96,15**	**99,05**	**96,63**	**98,99**
GLCM_MD_C	96,99	0,94	95,12	98,87	95,78	98,82
LBP_F [33]	95,21	**0,59**	94,03	96,39	94,22	96,25
HOG_C [36]	95,79	0,83	94,14	97,45	94,42	96,79
CTGF_3x3_F	94,44	1,03	92,38	96,50	93,17	96,59
CTGF_MDMS_F	94,31	1,40	91,51	97,10	91,48	95,74
GLCM_F [3,4]	93,62	1,13	91,36	95,89	91,82	95,23
GLCM_C [3,4]	94,19	1,36	91,47	96,91	92,24	96,45
CTGF_GLCM_MDMS_D	91,81	1,47	88,86	94,76	89,38	94,42
LBP_C [33]	90,20	1,22	87,77	92,64	88,16	91,71
CTGF_3x3_D	90,44	1,35	87,73	93,15	88,03	93,26
GLCM_MD_D	89,31	***2,28***	84,75	93,87	84,32	92,72
GLCM_MDMS_D	88,58	1,58	85,43	91,74	86,34	90,69
LBP_D [33]	88,08	1,50	85,08	91,07	86,17	90,19
CTGF_5x5_F	87,78	1,67	84,44	91,11	85,32	90,46
RECONST_ERROR_C [7]	84,87	2,09	80,69	89,04	81,79	87,82
CTGF_MDMS_D	88,45	1,38	85,69	91,20	86,51	90,36
CTGF_7x7_F	83,80	2,09	79,62	87,98	80,89	86,50
CTGF_5x5_D	84,80	1,38	82,04	87,56	82,29	87,02
CTGF_7x7_D	83,85	1,94	79,97	87,73	81,56	88,36
GLCM_D [3,4]	82,57	***2,68***	77,21	87,93	78,75	87,31
HOG_D [36]	79,66	1,37	76,92	82,41	78,00	81,90
HOG_F [36]	74,36	0,81	72,74	75,97	72,74	75,47
NOISE_STATS_C [18]	68,87	1,10	66,68	71,06	67,29	70,66
DWT_STATS_D [22]	***36,57***	1,94	***32,68***	***40,46***	***33,00***	***39,93***
DWT_STATS_F [22]	***34,34***	1,81	***30,73***	***37,95***	***32,08***	***37,65***
NOISE_STATS_F [18]	42,27	1,12	40,03	44,51	40,48	43,76
NOISE_STATS_D [18]	39,82	1,50	36,82	42,82	37,27	42,13
DWT_STATS_C [22]	***28,70***	***3,19***	***22,32***	***35,07***	***24,96***	***33,56***

Legend:
xx.xx = Three best methods in the column metric
xx.xx = Three worst methods in the column metric

In this paper, we propose to look beyond these simple texture approaches and analyze the multidirectional and multiscale properties of textures from printed documents. As Fig. 2 depicts (Section 4.3), by investigating printed letters in a microscope, we can see that the texture is spread over multiple directions. Hence, as expected, the GLCM_MD showed good classification results, 96.99% for characters, 97.15% for frames and 89.30% for documents. In addition, when considering the multidirectional and multiscale properties of texture patterns at the same time, GLCM_MDMS, the method yields the best result for characters: 97.60%. For frames, it also yielded a very good classification accuracy: 98.38%. For documents, it yielded an accuracy of 88.58%.

The proposed CTGF approaches were used here with filter sizes of 3×3, 5×5 and 7×7. These filters, when used individually, analyze the histograms of textures of low-level gradients. These textures are calculated on a neighborhood given by the filter size. These descriptors can be regarded as multidirectional filters. The CTGF with 3×3 filter size yielded accuracies of 94.44% for frames and 83.78% for documents. The 5×5 CTGF filter size yielded accuracies of 87.77% for frames and 80.28% for documents. Finally, the 7×7 CTGF filter size yielded accuracies of 83.80% for frames and 76.90% for documents. The multidirectional and multiscale approach in CTGF results in accuracies of 94.19% for frames and 88.45% for documents.

The fusion of CTGF with the GLCM uses the complementarity of both techniques. We combined the best proposed CTGF technique (CTGF_3x3) and the best multidirectional and multiscale technique (GLCM_MDMS). This last technique better explores the printing patterns more apparent between the printed material and background while CTGF explores micro-textures in regions of low gradient. This fusion yielded the best result of the experiment: a remarkable 98.47% classification accuracy for Frames. We also tried this fusion considering the entire document other than on frames it was not as effective: 91.81%.

Our second discussion on the experiments results is about how the techniques behave on the classification for each printer. For that, we show on Table 4 the f-measure as percentages.

As Table 4 shows, the multidirectional approach used by LBP is useful to identify the texture patterns of printer B4070 in frames, showing an f-measure of 100%. The voting approach and high

Table 4
F-measure of each technique per printer. The proposed techniques in this paper are the ones in bold in the column "Methods".

Method	Mean f-measure by Printer on Crossfolding 5x2 Experiments									
	B4070	C1150	C3240	C4370	H1518	H225A	H225B	LE260	OC330	SC315
	1	2	3	4	5	6	7	8	9	10
CTGF_GLCM_MDMS_F	99,76	98,78	98,82	99,57	98,90	***94,59***	***94,76***	99,12	**100,00**	**99,92**
GLCM_MDMS_F	99,24	99,04	99,16	99,24	98,73	***94,22***	***94,45***	99,41	**100,00**	**99,92**
GLCM_MD_F	98,51	98,08	97,44	97,90	98,89	***90,22***	***90,70***	99,25	**100,00**	**99,92**
GLCM_MDMS_C	99,59	95,45	99,01	99,03	94,67	***94,36***	***94,19***	**99,68**	**100,00**	99,59
GLCM_MD_C	99,60	95,02	97,95	97,59	94,14	***92,69***	***92,92***	**100,00**	**100,00**	99,51
LBP_F [33]	**100,00**	97,25	98,15	99,41	97,22	***82,06***	***77,46***	99,21	**100,00**	99,67
HOG_C [36]	95,24	92,63	97,44	98,28	93,74	***91,58***	***91,33***	97,41	**100,00**	**99,51**
CTGF_3x3_F	97,85	96,51	89,59	91,90	95,35	***87,89***	***89,31***	96,23	**99,65**	**99,50**
CTGF_MDMS_F	98,08	93,43	93,20	93,65	96,43	***84,72***	***88,11***	96,78	**98,64**	**99,59**
GLCM_F [3,4]	97,23	87,65	91,95	96,13	95,32	***84,12***	***86,44***	97,58	**99,17**	**99,92**
GLCM_C [3,4]	97,90	***89,55***	90,54	94,94	93,35	***88,27***	90,69	96,26	**100,00**	**99,58**
CTGF_GLCM_MDMS_D	**97,30**	89,65	88,87	92,47	96,22	***82,87***	***82,25***	93,82	95,66	**98,02**
LBP_C [33]	98,86	92,22	94,50	95,94	93,61	***73,84***	***49,39***	94,71	**99,83**	**99,43**
CTGF_3x3_D	91,48	90,26	87,40	***85,20***	92,30	88,21	***85,95***	89,98	**95,48**	**97,83**
GLCM_MD_D	**95,85**	88,78	91,32	88,69	94,17	***76,51***	***75,85***	90,71	94,16	**95,54**
GLCM_MDMS_D	92,79	83,88	88,29	91,02	**93,83**	***76,51***	***78,11***	93,54	90,25	**96,32**
LBP_D [33]	92,82	87,24	87,87	90,23	93,96	***72,68***	***71,18***	91,02	**94,79**	**97,40**
CTGF_5x5_F	87,47	83,42	84,30	81,59	93,20	***77,90***	***78,37***	94,26	**97,13**	**98,81**
RECONST_ERROR_C [7]	87,43	90,75	90,34	92,74	92,47	***43,72***	***48,11***	95,18	**98,01**	**97,96**
CTGF_MDMS_D	91,25	84,30	***83,59***	88,49	88,15	84,53	***83,37***	91,53	**91,67**	**96,44**
CTGF_7x7_F	85,46	78,89	***69,58***	83,64	93,48	***71,14***	74,18	88,04	**96,48**	**97,42**
CTGF_5x5_D	87,13	***75,73***	80,58	82,92	90,51	***75,70***	77,71	90,94	**91,95**	**94,36**
CTGF_7x7_D	86,38	***74,30***	***75,44***	82,78	**89,27**	81,90	80,19	83,86	88,58	**95,70**
GLCM_D [3,4]	**93,90**	73,33	81,89	76,77	92,81	***71,86***	***69,03***	85,95	85,82	**93,61**
HOG_D [36]	85,41	71,43	81,59	81,14	**92,31**	***54,01***	***53,90***	89,78	91,79	**93,37**
HOG_F [36]	77,57	64,18	71,90	68,28	**94,05**	***51,20***	***46,60***	86,87	**92,70**	86,09
NOISE_STATS_C [18]	***45,21***	54,60	***32,56***	57,71	92,88	69,81	48,99	72,49	**93,07**	**96,42**
DWT_STATS_D [22]	15,01	***12,27***	21,78	21,40	**92,60**	39,74	***10,23***	28,08	37,64	**53,93**
DWT_STATS_F [22]	19,32	15,16	***0,65***	14,68	**94,24**	34,31	***0,00***	42,53	15,58	**43,94**
NOISE_STATS_F [18]	55,04	18,94	***1,87***	38,65	**77,15**	***11,40***	40,39	18,47	35,56	**59,59**
NOISE_STATS_D [18]	38,01	27,25	51,94	38,93	**67,51**	26,01	31,02	***20,75***	***18,44***	**75,09**
DWT_STATS_C [22]	29,19	2,00	***0,00***	0,30	**93,82**	3,96	***0,00***	8,16	**56,24**	25,60

Legend:
xx.xx = Two best *f-measure* results of the method
xx.xx = Two worst f-measure results of the method

Table 5
Confusion matrix for the best proposed technique: the fusion of the convolution texture gradient filter with 3 × 3 mask and multidirectional and multiscale GLCMs applied on frames (CTGF_GLCM_MDMS_F). Results shown are in percentages.

CTGF_GLCM_MDMS_F			Attributed Printer ID									
			B4070	C1150	C3240	C4370	H1518	H225A	H225B	LE260	OC330	SC315
			1	2	3	4	5	6	7	8	9	10
Actual Printer ID	B4070	1	99.66	0.34								
	C1150	2	0.18	99.82								
	C3240	3			100.00							
	C4370	4			0.84	99.16						
	H1518	5		1.34			97.82	0.84				
	H225A	6		0.34				94.92	4.75			
	H225B	7						5.32	94.68			
	LE260	8			1.53					98.47		
	OC330	9									100.00	
	SC315	10		0.17								99.83

presence of texture in the printed material explain the high f-measure for this technique for that printer. The f-measure from the proposed multidirectional and multiscale approaches (GLCM_MDMS and GLCM_MD) and the fusion of CTGF and GLCM_MDMS (CTGF_GLCM_MDMS) also present a high f-measure for this printer. Hence, texture micro patterns explored by these techniques are more important to identify this printer than its noise signature, explained by the very low f-measure of NOISE_STATS (45.21%). Another multidirectional approach proposed in this paper, the GLCM_MD, shows the best f-measure to identify printer LE260 (100%).

For printer C1150, the multidirectional approach is not totally discriminative by itself. The highest f-measure for this task was achieved by the proposed multidirectional and multiscale GLCM on characters – GLCM_MDMS_C, with an f-measure of 99.04%. The same behavior is noticed on printers C3240, C4370, H1518, H225A, H225B. The multidirectional and multiscale texture micro patterns analyses and also the fusion proposed in this paper showed the best f-measures (99.16, 99.57, 98.90, 94.59 and 94.76%, respectively). Note also that the proposed methods performed well on printers H225A and H225B, which are from the same manufacturer and model, but with few firmware modifications. These are the most difficult printers to identify (see Table 4).

Printers OC330 and SC315 are easy to identify. Both the multidirectional and multiscale GLCM and fusion proposed herein showed an 100% f-measure for printer OC330. The multidirectional HOG, LBP and the state-of-the-art GLCM also showed an 100% f-measure for printer OC330. Therefore, any of these techniques is enough to identify this printer. For printer SC315, both the multidirectional and multiscale GLCM and the CTGF_GLCM_MDMS_F fusion yielded 99.62% of f-measure. The multidirectional GLCM and the state-of the-art GLCM yielded the same f-measure. Tables 5 and 6 show the confusion matrices for the two best proposed techniques using percentages.

Finally, we also performed statistical tests to compare all the techniques. Performing a Friedmann test on f-measure values yielded a p-value of 2.2262×10^{-203}, indicating that there is statistical difference among the techniques. This allows us to perform the Tukey-HSD pairwise tests. Table 7 shows such results using the 15 best methods considering the f-measure. Each (*line*, *column*) in this table shows whether a technique wins (*line*), loses or draw in the statistical analysis against another technique (*column*). The higher the method is in the rows, the better. A Bonferroni statistical test in this analysis also confirms the findings.

There are at least two interest aspects to observe in Table 7. The first one is regarding some of the proposed methods (GLCM_MDMS, CTGF_GLCM_MDMS and GLCM_MD). The techniques in this cluster are not statistically significant when compared pairwise with other members of the cluster. Interestingly, they explore multiscale and multidirectionality analyses we hypothesized as important for printer attribution confirming the hypothesis. Also, the two CTGF_GLCM_MDMS and GLCM_MDMS are the two best performing techniques. The second one regard the cluster comprising the state-of-the-art techniques for printer attribution, the two general-purpose texture descriptors (LBP and HOG) and one variation of the proposed methods (CTGF 3 × 3 on frames and documents). In this cluster, most of techniques are statistically significant from most of others in the same group.

From Table 7, we can see that the multiscale and multidirectional analyses proposed for frames of documents with sufficient printed material is more effective than the state-of-the-art

Table 6
Confusion matrix for the second best proposed technique: the gray-level co-occurrence matrices multidirectional and multiscale on frames (GLCM_MDMS_F). Results shown are in percentages.

GLCM_MDMS_F			Attributed Printer ID									
			B4070	C1150	C3240	C4370	H1518	H225A	H225B	LE260	OC330	SC315
			1	2	3	4	5	6	7	8	9	10
Actual Printer ID	B4070	1	100.00									
	C1150	2	0.35	99.47			0.18					
	C3240	3			98.99	1.01						
	C4370	4			0.50	99.50						
	H1518	5	0.34	1.18			97.65	0.84				
	H225A	6						92.54	7.46			
	H225B	7						3.30	96.70			
	LE260	8	0.85		0.17					98.98		
	OC330	9									100.00	
	SC315	10		0.17								99.83

Table 7
Tukey-HSD pairwise statistical test results using f-measures of the 15 best methods present in Table 3. The value 0 means that there is no statistical difference between the methods. The value 1 means that the method in the corresponding row is statistically better than the method in the corresponding column while −1 means otherwise.

Method	CTGF_GLCM_MDMS_F	GLCM_MDMS_F	GLCM_MDMS_C	GLCM_MD_F	GLCM_MD_C	LBP_F [33]	HOG_C [36]	CTGF_3x3_F	GLCM_C [3,4]	GLCM_F [3,4]	CTGF_GLCM_MDMS_D	CTGF_3x3_D	RECONST_ERROR_C [7]	NOISE_STATS_C [18]	DWT_STATS_D [22]
CTGF_GLCM_MDMS_F	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
GLCM_MDMS_F	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
GLCM_MDMS_C	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
GLCM_MD_F	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
GLCM_MD_C	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
LBP_F [33]	-1	-1	0	0	0	0	0	1	1	1	1	1	1	1	1
HOG_C [36]	-1	-1	-1	-1	0	0	0	0	0	0	1	1	1	1	1
CTGF_3x3_F	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1	1	1	1
GLCM_C [3,4]	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1	1	1	1
GLCM_F [3,4]	-1	-1	-1	-1	-1	-1	0	0	0	0	0	1	1	1	1
CTGF_GLCM_MDMS_D	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1
CTGF_3x3_D	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	1	1
RECONST_ERROR_C [7]	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1
NOISE_STATS_C [18]	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
DWT_STATS_D [22]	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0

1 = Line method is better than column method
0 = Line method is equivalent to column method
-1 = Line method is worse than column method

techniques for the dataset we consider herein (with different letter sizes, styles and figures). This is noticeable by the two best techniques ranked in that table, CTGF_GLCM_MDMS_F and GLCM_MDMS_F.

7. Conclusion

In this paper, we propose new approaches for laser printer attribution, a problem of paramount importance because they can be a powerful tool to help solving crimes involving documents. Our solutions work beyond simple texture approaches as they analyze how the texture on printed text and figures behave when using multidirectional and multiscale analysis.

Our first contribution to achieve this task are descriptors based on statistics of the multidirectional gray-level co-occurrence matrices (GLCM_MD) and multidirectional and multiscale gray-level co-occurrence matrices (GLCM_MDMS). The first one analyzes multiple neighboring directions and the second analyzes multiple neighboring directions in a pyramidal Gaussian image decomposition. This can be helpful when texture spreads over multiple directions and scales. Our second contribution is the convolution gradient texture filter, which considers low-gradient micro-texture patterns. This descriptor is also multidirectional as it calculates textures over a neighborhood with different kernel sizes. It analyzes the frequency of how a pixel relates to its neighbors on areas of low-level gradients (*i.e.*, inside printing area) in texts and figures printed by different printers. We also proposed a fusion between the analyses made by both of them.

Our last contribution refers to the best place to look at on printed documents to better investigate printing patterns. By analyzing areas of text with enough printing material, we can identify the laser source printer in a better way than just looking at characters and documents as more printing textures and less background are available. This technique has the same advantage of the characters analysis, that is representing a document with multiple feature vectors, classifying them individually and fusing the individual classifications in the end. An additional advantage when compared to a full-document analysis is that this method can be applied if just parts of the document are available.

We compared the proposed approaches against some state-of-the-art and some general purpose texture descriptors in Wikipedia scanned documents and showed their effectiveness when the characterization occurs in characters, frames and documents. The techniques proposed herein yielded the first and second best classification accuracies when applied on the proposed frames. They were the best to identify 90% of the printers and results are statistically different when compared with the state-of-the-art counterparts. The take-home lesson is that the multidirectional analysis is crucial for laser printer attribution, specially when combined with multiscale image decomposition.

From our experience, it is important to highlight that laser printer attribution is a very difficult problem in which many variables play a role. First of all, the reference scanner used in the scanning process when defining the training samples and analyzing an investigated document must be the same as we do not want intrinsic scanning features to play a key role in the printer

attribution problem. When using the same scanner for training and investigated documents, we rule out this effect. The scanning process inserts intrinsic features in the documents, which can be used to identify the scanning device. This is known as Scanner Attribution in the literature and there are very good work on this as references [42,43] show.

This is not a major problem for the forensic expert because our application here is to identify the printer source of a document. So, the scanner variable can be fixed. There are some situations where different scanners have very similar variables (resolution and noise), but we cannot guarantee that in all practical scenarios. Therefore, we recommend that the scanner used for acquiring the investigated documents should be the same as the one used for training the classifier. As just a few documents are necessary for training the classifiers, this is straightforward. This procedure is also used in other devices attribution (cameras and scanners). When a suspect camera is investigated, the classifier must be retrained with data acquired with that camera [44–46].

Second, it is advisable to use, as much as possible, similar paper to the one collected for investigation. If the investigated document for printer attribution is a white office Letter with 75 g/m^2, it is recommended to use a similar paper in the training (acquisition of training documents from the suspect printers). If we use training data considering photographic reflective paper, for instance, which are very different from the investigated printed document, it is likely the proposed methods and their counterparts in the literature using vision-based approaches will fail.

In addition, the good results presented in this paper must come with a salt of grain as well. We are not claiming to have solved the printer attribution problem. The almost 99% classification accuracy is an important and unrivaled result. However, each real case will have its specificities. The forensic expert must be aware that vision-based approaches are an initial, non-destructive and cheap analysis. However, it must be used, whenever possible, with other techniques in order to provide the most accurate results as possible. The vision-based techniques can also be combined to improve the quality of the attribution.

Finally, we envision at least two research paths for extending this research. First, an in-depth study of the analyzed techniques on color documents with proper adaption of the methods for this scenario is worth exploring. Second, an investigation of more complementarity approaches of the methods proposed would be interesting to check if classifier and decision-level fusion could push the classification results even further.

Acknowledgements

We thank the financial support of CNPq (Grants #477662/2013-7, and #304352/2012-8), FAPESP (Grant #2010/05647-4), the CAPES DeepEyes project, and Microsoft Research. In addition, we also thank to Dr. Elizabeth S. Chen and Federal University of S ao Paulo, who kindly assisted us with the microscope analysis of the printed letters.

Appendix A. Gray-level co-occurrence matrices features

The work of Miklineni et al. [3] proposed a set of features calculated on top of gray-level co-occurrence matrices. We use this same set of features in this work.

Before the features are calculated a set of definitions are extracted from the image: (1) Number of pixels R in a Region of Interest (ROI), which is the set of all pixels within the printed area of the character; (2) The gray-level co-occurrence matrices $glcm(n, m)$, which are two-dimensional histograms per neighborhood direction (dr, dc) showing the occurrence of pixels n and m in a given distance (dr, dc); (3) The number of neighboring ROI pixels distant by a (dr, dc) offset R_{glcm}; (4) GLCM probability estimates p_{glcm}; (5) marginal probability densities in the row and column directions p_r and p_c; (6) histograms of differences $D(k)$; (7) histograms of sums $S(k)$ and its mean μ_S; (8) Mean pixel of a ROI and (9) density of a ROI. Eqs. (A.1)–(A.11) formalize these calculations.

$$R = \sum_{(i,j) \in ROI} 1 \tag{A.1}$$

$$glcm(n,m) = \sum_{(i,j),(i+dr,j+dc) \in ROI} 1_{\{I(i,j)=n, I(i+dr,j+dc)=m\}} \tag{A.2}$$

$$R_{glcm} = \sum_{(i,j),(i+dr,j+dc) \in ROI} 1 \tag{A.3}$$

$$p_{glcm}(n,m) = \frac{1}{R_{glcm}} glcm(n,m) \tag{A.4}$$

$$p_r(n) = \sum_{m=0}^{255} p_{glcm}(n,m) \tag{A.5}$$

$$p_c(m) = \sum_{n=0}^{255} p_{glcm}(n,m) \tag{A.6}$$

$$D(k) = \sum_{\substack{0 \le n \le 255 \\ 0 \le m \le 255 \\ |n-m| = k}} p_{glcm}(n,m) \tag{A.7}$$

$$S(k) = \sum_{\substack{0 \le n \le 255 \\ 0 \le m \le 255 \\ n+m = k}} p_{glcm}(n,m) \tag{A.8}$$

$$\mu_S = \sum_{k=0}^{510} k S(k) \tag{A.9}$$

$$\mu_{ROI} = \frac{1}{R} \sum_{(i,j) \in ROI} I(i,j) \tag{A.10}$$

$$p_{ROI}(k) = \frac{1}{R} 1_{\{I(i,j)=k\}} \tag{A.11}$$

Eleven features are calculated from the data in Eqs. (A.1)–(A.6). The first four are marginal means and variances defined in Eqs. (A.12)–(A.15).

$$\mu_r = \sum_{n=0}^{255} n\, p_r(n) \tag{A.12}$$

$$\mu_c = \sum_{m=0}^{255} m\, p_c(m) \tag{A.13}$$

$$\sigma_r^2 = \sum_{n=0}^{255} n^2\, p_r(n) - \mu_r^2 \tag{A.14}$$

$$\sigma_c^2 = \sum_{m=0}^{255} m^2\, p_c(m) - \mu_c^2 \tag{A.15}$$

The next seven features are the energy of the normalized GLCM, three entropy measurements, the maximum entry in the GLCM,

and two correlation metrics.

$$E = \sum_{n=0}^{255}\sum_{m=0}^{255} p_{glcm}^2(n,m) \tag{A.16}$$

$$H_{rc1} = -\sum_{n=0}^{255}\sum_{m=0}^{255} p_{glcm}(n,m)\log_2(p_r(n)\,p_c(m)) \tag{A.17}$$

$$H_{rc2} = -\sum_{n=0}^{255}\sum_{m=0}^{255} p_r(n)\,p_c(m)\log_2(p_r(n)\,p_c(m)) \tag{A.18}$$

$$H_{glcm} = -\sum_{n=0}^{255}\sum_{m=0}^{255} p_{glcm}(n,m)\log_2(p_{glcm}(n,m)) \tag{A.19}$$

$$P_{\max} = \max\{p_{glcm(n,m)}\} \tag{A.20}$$

$$\rho_1 = \sum_{n=0}^{255}\sum_{m=0}^{255} \frac{(n-\mu_r)(m-\mu_c)\,p_{glcm}(n,m)}{\sigma_r\sigma_c} \tag{A.21}$$

$$\rho_2 = \sum_{n=0}^{255}\sum_{m=0}^{255} |n-m|(n+m-\mu_r-\mu_c)\,p_{glcm}(n,m) \tag{A.22}$$

Four features, Eqs. (A.23)–(A.26), are obtained from the difference histogram $D(k)$ defined by Eq. (A.7). They are the energy, entropy, inertia, and local homogeneity of $D(k)$ respectively.

$$E_D = \sum_{k=0}^{255} D^2(k) \tag{A.23}$$

$$H_D = -\sum_{k=0}^{255} D(k)\log_2 D(k) \tag{A.24}$$

$$I_D = \sum_{k=0}^{255} k^2 D(k) \tag{A.25}$$

$$h_D = \sum_{k=0}^{255} \frac{D(k)}{1+k^2} \tag{A.26}$$

Five features, Eqs. (A.27)–(A.31), are obtained from the sum $S(k)$ histogram defined by Eqs. (A.8) and (A.9). They are the energy, entropy, variance, cluster shade, and cluster prominence of $S(k)$, respectively.

$$E_S = \sum_{k=0}^{510} S^2(k) \tag{A.27}$$

$$H_S = -\sum_{k=0}^{510} S(k)\log_2 S(k) \tag{A.28}$$

$$sigma_S^2 = \sum_{k=0}^{510} (k-\mu_S)^2 S(k) \tag{A.29}$$

$$A = \sum_{k=0}^{510} \frac{(k-\mu_r-\mu_c)^3 S(k)}{(\sigma_r^2+\sigma_c^2+2\sigma_r\sigma_c)^{3/2}} \tag{A.30}$$

$$B = \sum_{k=0}^{510} \frac{(k-\mu_r-\mu_c)^4 S(k)}{(\sigma_r^2+\sigma_c^2+2\sigma_r\sigma_c)^{2}} \tag{A.31}$$

The last two features use data in Eqs. (A.10) and (A.11). These are the ROIs variance and entropy, as shown in Eqs. (A.32) and (A.33).

$$sigma_{ROI}^2 = \frac{1}{R}\sum_{(i,j)\in ROI} (I(i,j)-\mu_{ROI})^2 \tag{A.32}$$

$$H_{ROI} = -\sum_{k=0}^{255} p_{ROI}(k)\log_2 p_{ROI}(k), \tag{A.33}$$

which completes the set of 22 GLCM features considered.

References

[1] P.-J. Chiang, N. Khanna, A. Mikkilineni, M. Segovia, S. Suh, J. Allebach, G.-C. Chiu, E. Delp, Printer and scanner forensics, Signal Process. Mag. 26 (2) (2009) 72–83.
[2] G.N. Ali, P. ju Chiang, A.K. Mikkilineni, G.T. Chiu, E.J. Delp, J.P. Allebach, Application of principal components analysis and Gaussian mixture models to printer identification, in: Intl. Conference on Digital Printing Technologies, 2004, 301–305.
[3] A.K. Mikkilineni, P. ju Chiang, G.N. Ali, G.T.-C. Chiu, J.P. Allebach, E.J. Delp, Printer identification based on graylevel co-occurrence features for security and forensic applications, in: Intl. Conference on Security, Steganography and Watermarking of Multimedia Contents, 2005, 430–440.
[4] A.K. Mikkilineni, P. ju Chiang, G.N. Ali, G.T. Chiu, J.P. Allebach, E.J. Delp, Printer identification based on textural features, in: Intl. Conference on Digital Printing Technologies, 2004, 306–311.
[5] A.K. Mikkilineni, O. Arslan, P. ju Chiang, R.M. Kumontoy, J.P. Allebach, G.T. Chiu, Printer forensics using SVM techniques, in: Intl. Conference on Digital Printing Technologies, 2005, 223–226.
[6] M.-J. Tsai, J.-S. Yin, I. Yuadi, J. Liu, Digital forensics of printed source identification for Chinese characters, Multimed. Tools Appl. 73 (2013) 1–27.
[7] E. Kee, H. Farid, Printer profiling for forensics and ballistics, in: ACM Workshop on Multimedia and Security, 2008, 3–10.
[8] R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial greyscale, Soc. Inf. Disp. 17 (2) (1976) 75–77.
[9] R. Ulichney, Digital Halftoning, MIT Press, Cambridge, MA, 1987.
[10] N. Khanna, A.K. Mikkilineni, G.T. Chiu, J.P. Allebach, E.J. Delp, Survey of scanner and printer forensics at Purdue University, in: Intl. Workshop on Computational Forensics, 2008, 22–34.
[11] G.N. Ali, P.-J. Chiang, A.K. Mikkilineni, J.P. Allebach, G.T.-C. Chiu, E.J. Delp, Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices, in: Intl. Conference on Digital Printing Technologies, 2003, 511–515.
[12] J.E. Girard, Criminalistics: Forensic Science, Crime and Terrorism, 3rd ed., Jones and Bartlett Publishers, Burlington, MA, 2013.
[13] A. Braz, M. Lopez-Lopez, C. Garcia-Ruiz, Raman spectroscopy for forensic analysis of inks in questioned documents, Forensic Sci. Int. 232 (1–3) (2013) 206–212.
[14] L. Gal, M. Belovicova, M. Oravec, M. Palkova, M. Ceppan, Analysis of laser and inkjet prints using spectroscopic methods for forensic identification of questioned documents, in: Symposium on Graphic Arts, vol. 10, 1993, 1–8.
[15] P.-C. Chu, B.Y. Cai, Y.K. Tsoi, R. Yuen, K.S. Leung, N.-H. Cheung, Forensic analysis of laser printed ink by x-ray fluorescence and laser-excited plume fluorescence, Anal. Chem. 85 (9) (2013) 4311–4315.
[16] O. Bulan, J. Mao, G. Sharma, Geometric distortion signatures for printer identification, in: Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP), 2009, 1401–1404.
[17] Y. Wu, X. Kong, X. You, Y. Guo, Printer forensics based on page document's geometric distortion, in: Intl. Conference on Image Processing (ICIP), 2009, 2909–2912.
[18] A.K. Mikkilineni, N. Khanna, E.J. Delp, Forensic printer detection using intrinsic signatures, in: Intl. Society for Optics and Photonics (SPIE), vol. 7880, 78800R-78800R-11, 2011.
[19] H.-Y. Lee, J.-H. Choi, Identifying color laser printer using noisy feature and support vector machine, in: Intl. Conference on Ubiquitous Information Technologies and Applications, 2010, 1–6.
[20] J.-H. Choi, H.-K. Lee, H.-Y. Lee, Y.-H. Suh, Color laser printer forensics with noise texture analysis, in: ACM Workshop on Multimedia and Security, 2010, 19–24.
[21] S. Elkasrawi, F. Shafait, Printer identification using supervised learning for document forgery detection, in: Intl. Workshop on Document Analysis Systems, 2014, 146–150.
[22] M.U. Devi, C.R. Rao, A. Agarwal, A survey of image processing techniques for identification of printing technology in document forensic perspective, Int. J. Comput. Appl. 1 (2010) 9–15.
[23] N. Khanna, A.K. Mikkilineni, A.F. Martone, G.N. Ali, G.T.-C. Chiu, J.P. Allebach, E.J. Delp, A survey of forensic characterization methods for physical devices, Digit. Investig. 3 (2006) 17–28.
[24] R. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, Trans. Syst. Man Cybern. SMC-3 (6) (1973) 610–621.
[25] J.-H. Choi, D.-H. Im, H.-Y. Lee, J.-T. Oh, J.-H. Ryu, H.-K. Lee, Color laser printer identification by analyzing statistical features on discrete wavelet transform, in: Intl. Conference on Image Processing (ICIP), 2009, 1505–1508.
[26] M.-J. Tsai, J. Liu, C.-S. Wang, C.-H. Chuang, Source color laser printer identification using discrete wavelet transform and feature selection algorithms, in: Intl. Symposium on Circuits and Systems (ISCAS), 2011, 2633–2636.
[27] W. Jiang, A.T.S. Ho, H. Treharne, Y.Q. Shi, A novel multi-size Block Benford's law scheme for printer identification, in: Pacific Rim Conference on Advances in Multimedia Information Processing, PCM'10, 2010, 643–652.
[28] S.-J. Ryu, H.-Y. Lee, D.-H. Im, J.-H. Choi, H.-K. Lee, Electrophotographic printer identification by halftone texture analysis, in: IEEE Intl. Conference on Acoustics Speech and Signal Processing (ICASSP), 2010, 1846–1849.

[29] J. Shi, J. Malik, Normalized cuts and image segmentation, Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[30] R. Duda, P. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, New York, 1973.
[31] M. Gaubatz, S. Simske, Printer-scanner identification via analysis of structured security deterrents, in: Intl. Workshop on Information Forensics and Security (WIFS), 2009, 151–155.
[32] S.J. Simske, J.S. Aronoff, M. Sturgill, J.C. Villa, Spectral pre-compensation and security print deterrent authentication, NIP Digit. Fabr. Conf. 2008 (2) (2008) 792–795.
[33] W. Mazzella, R. Marquis, Forensic image analysis of laser-printed documents, J. Am. Soc. Quest. Doc. Exam. 10 (1) (2007) 19–24.
[34] M. Schreyer, Intelligent printing technique recognition and photocopy detection for forensic document examination, in: L. Porada (Ed.), Informatiktage, vol. S-8, 2009, pp. 39–42.
[35] F.R. de Siqueira, W.R. Schwartz, H. Pedrini, Multi-scale gray level co-occurrence matrices for texture description, Neurocomputing 120 (2013) 336–345, Image Feature Detection and Description.
[36] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[37] T. Ojala, M. Pietikinen, D. Harwood, A comparative study of texture measures with classification based on feature distributions, Pattern Recognit. 29 (1996) 51–59.
[38] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Comput. 10 (1998) 1895–1923.
[39] N. Khanna, G.T.C. Chiu, J.P. Allebach, E.J. Delp, Scanner identification with extension to forgery detection, in: Intl. Society for Optics and Photonics (SPIE), vol. 6819, 68190G-68190G-10, 2008.
[40] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[41] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: C. Schmid, S. Soatto, C. Tomasi (Eds.), Intl. Conference on Computer Vision & Pattern Recognition, vol. 2, 2005, 886–893.
[42] P.-J. Chiang, N. Khanna, A.K. Mikkilineni, M.V.O. Segovia, S. Suh, J.P. Allebach, G.T.C. Chiu, E.J. Delp, Printer and scanner forensics, IEEE Signal Process. Mag. 26 (2) (2009) 72–83.
[43] N. Khanna, A.K. Mikkilineni, E.J. Delp, Scanner identification using feature-based processing and analysis, IEEE Trans . Inf. Forensics Secur. 4 (1) (2009) 123–139.
[44] M. Chen, J. Fridrich, M. Goljan, J. Lukas, Determining image origin and integrity using sensor noise, IEEE Trans. Inf. Forensics Secur. 3 (1) (2008) 74–90.
[45] J. Lukas, J. Fridrich, M. Goljan, Digital camera identification from sensor noise sensor, IEEE Trans. Inf. Forensics Secur. 1 (2) (2006) 205–214.
[46] F. de Oliveira Costa, E.A. Silva, M. Eckmann, W.J. Scheirer, A. Rocha, Open set source camera attribution and device linking, Pattern Recognit. Lett 39 (2014) 92–101, Elsevier.