EURASIP Journal on
Advances in Signal Processing
a SpringerOpen Journal

RESEARCH   **Open Access**

Detecting documents forged by printing and copying

Shize Shang[1], Nasir Memon[2] and Xiangwei Kong[1]*

Abstract

This paper describes a method to distinguish documents produced by laser printers, inkjet printers, and electrostatic copiers, three commonly used document creation devices. The proposed approach can distinguish between documents produced by these sources based on features extracted from the characters in the documents. Hence, it can also be used to detect tampered documents produced by a mixture of these sources. We analyze the characteristics associated with laser/inkjet printers and electrostatic copiers and determine the signatures created by the different physical and technical processes involved in each type of printing. Based on the analysis of these signatures, we computed the features of noise energy, contour roughness, and average gradient. To the best of our knowledge, this is the first work to distinguish documents produced by laser printer, inkjet printer, and copier based on features extracted from individual characters in the documents. Experimental results show that this method has an average accuracy of 90% and works with JPEG compression.

Keywords: Device type identification; Tampering detection; Noise energy; Contour roughness; Average gradient

1 Introduction

Even in the age of the Internet and digital communication, paper is used extensively as an important carrier of information. Technological advances, however, have made the printed document easy to modify for malicious purposes. Therefore, document authenticity plays an important role in forensic science where documents are disputed in a court of law. In fact, 'questioned document examination' (QDE) has become an important discipline within forensic science. QDE deals with scientific techniques that can provide evidence about a suspicious or questionable document.

Traditional techniques for document inspection depend on specialized devices, trained personnel, and chemicals. Such systems can be expensive, cause wear and tear, and the chemicals used can even cause damage to the documents. Document forensics in the digital domain evolved to alleviate the need for specialized devices and trained personnel in order to determine the integrity of documents being analyzed.

Document forensics technology, mostly focused on tracing the source of a document or on detecting forgery, has developed rapidly in recent years. This technology uses commodity scanners and a computer to perform the necessary analyses. The analyses can be automatic or semi-automatic, reducing costs while increasing convenience. Nevertheless, document forensics faces many challenges that limit its development. The techniques are currently limited to text documents with black text on white background. In addition, the device artifacts on the questioned document do not transfer without loss to the scanned image that is analyzed. Various methods in digital image forensics are reviewed in [1], edited by Sencar and Memon. Although document forensics works based on scanned images, the methods are significantly different. Recent research in document forensics has focused on printer identification [2-8], forgery detection [9-12], and scanner identification [13-16].

In this paper, we describe a method to distinguish documents produced by laser printers, inkjet printers, and electrostatic copiers, three devices that are commonly used today. The proposed approach can distinguish between documents produced by these sources based on the characters in the document. Hence, it can also

*Correspondence: kongxw@dlut.edu.cn
[1]Dalian University of Technology, No. 2 Linggong Road, Ganjingzi District, Dalian 116024, China
Full list of author information is available at the end of the article

be used to detect tampered documents produced by a mixture of these sources. We analyze the characteristic for laser/inkjet printers and electrostatic copiers and determine the telltale 'signatures' caused by the different physical and technical processes involved in each type of printing. Based on the analysis of these signatures, we compute features comprising noise energy, contour roughness, and average gradient. To the best of our knowledge, this is the first work to distinguish documents produced by laser printers, inkjet printers, and copiers based on features extracted from individual characters in the documents. Using individual characters also allows us to detect and localize document forgeries created using different types of source documents.

This paper is organized as follows: Section 2 introduces related work in print forensics. Section 3 reviews the working principles of printers and copiers and the different characteristics they impart on documents. Section 4 describes the features used by the proposed technique. Section 5 gives the experimental results of the proposed scheme, including a description of how the different features perform, the performance of the algorithm for JPEG compression, and different resolutions; and Section 6 presents conclusions and future work.

2 Related work

Often, the first step in document forensics focuses on source identification. The aim is to trace the source of the documents including the brand and model of the device. Typically, statistical characteristics of the printed characters are used to identify the source brands and models. For example, Delp et al. used gray-level co-occurrence matrices (GLCM) [2,3] features, variance/entropy, and discrete Fourier transform (DFT) [4] features to identify the printer source. Banding frequency is another characteristic used to identify the source as a laser printer. Banding, which appears as non-uniform light and dark lines perpendicular to the printing direction [5], is caused by fluctuations in the rotating angular velocity of the organic photo conductor (OPC) drum and by errors in the gear transmission mechanism. Delp et al. [6] printed pre-generated gray images and exploited the projection method and Fourier transform to estimate banding frequencies. Some researchers have also focused on hardware defects to trace the source; these include spinning velocity fluctuations of the polygon mirror in the laser printer and the imperfections of the paper feeding mechanism, which result in geometric distortions in the document that can serve to characterize individual laser printers. Kong et al. [7] established a projective transformation model to estimate such geometric distortions. Bulan et al. [8] computed the geometric distortion signature from printed halftone images to trace the source laser printer by estimating the variations in the center positions of halftone dots.

The second step in document forensics is document forgery detection. A forgery could involve changing, adding, or deleting some information on the document or replacing an entire page with a counterfeited page [9]. For pasting and reprinting forgery operations, character location distortion is often introduced. Beusekon et al. [10] presented a technique for extracting text lines and alignment lines for document inspection. For English language characters, most characters align according to the ascender line, descender line, and base line. Tampered characters deviate from these three lines because of location distortion. In [11], the matching quality between all pairs of documents was used to expose tampered documents. When a page is replaced or reprinted, location distortion will occur when comparing the forged page or the tampered region with a genuine document. By computing the matching quality of two page images, the forged page or tampered region will be detected. Farid and Kee [12] established a printer model for characters to detect documents forged by different printers. They used principal component analysis (PCA) and singular value decomposition (SVD) to model the degradation of a page caused by printing, and the resulting printer profile was then used to distinguish between characters generated from different printers.

If a technique that aims to detect the source device type (printer, copier, etc.) is based on characters, it can also be applied for forgery detection. Some work has been done on this topic. Chan [17,18] extracted the edge sharpness, surface roughness, and image contrast features from pre-printed images of squares. A neural network was then applied to distinguish print and copying techniques. Lampert and Breuel [19] analyzed the differences between laser and inkjet printouts; a set of 15 features was extracted from each character including line edge roughness, correlation coefficient, and texture. A support vector machine (SVM) was then used to classify the characters. The average accuracy for this technique reached 94.8%. Umadevi et al. [20] divided a text image into three regions: foreground text, noise, and background. An expectation maximization (EM) algorithm was then utilized to determine the three regions. An iterative algorithm was applied to generate a parameter print index (PI) used for print technology discrimination. All the methods in [17-20] focused on distinguishing characters produced by laser and inkjet printers. Schulze et al. [21] described a frequency domain method to distinguish between printed and copied documents. The mean and standard deviation of the discrete cosine transformation (DCT) coefficients were extracted from image blocks, and an average accuracy of 99.08% was achieved for full-page detection.

3 Characteristics of documents produced by different devices

3.1 The printing/copying process

In this section, we describe how printers/copiers work, and the characteristic telltale 'signatures' they generate in a document. Differing technical processes and mechanical constructions cause different character morphologies in the document. In the electrophotographic process of laser printing, there are six steps: charging, exposure, developing, transferring, fusing, and cleaning [5]. The optoelectronic devices in a laser printer accurately transfer the image signal corresponding to the document, and the toner image is melted by the fuser and pressed onto the paper. As a result, the printed characters have a glossy appearance and a clear contour.

An electrostatic copier utilizes the same electrophotographic process as a laser printer. However, an electrostatic copier scans the document using CCD/CMOS image sensors, converts the analog signal to a digital signal, and then performs the electrophotographic process.

An inkjet printer consists of three principal components: the print head, the carriage, and the paper-advance mechanism [5]. The print head is fixed to the carriage and fires ink onto paper while the carriage moves back and forth in the scan direction.

3.2 Differences in document characteristics

3.2.1 Laser printer vs. electrostatic copier

In contrast to the laser printer, the electrostatic copier has the additional step of scanning. In this step, significant degradation in quality is introduced to the characters (Figure 1). The initial degradation occurs when the document is first printed; character morphology is degraded from a perfect electronic signal to an analog printout. The next degradation occurs during document scanning. The dust and scratches on the scanner platen may degrade the quality, and sensor noise may also degrade the document quality. The copied characters thus have more additive white Gaussian noise (AWGN) and impulse noise than laser-printed characters.

3.2.2 Inkjet printer vs. laser printer

The primary difference between inkjet and laser printers lies in the process. An inkjet printer works by firing drops of ink onto paper when the print head is moving. As a result, tails or satellites of the ink drop are formed on the document, and the contours of the printed characters are rough and contain ups and downs. For some brands and models, the period of the ups and downs can be observed in the character contour because the distance between two scanning traces is constant for a printer, and the distance between two tails is therefore the same. In addition, the ink is a fluid that contains a considerable amount of water. The diffusion speed of the ink is quicker than that of the fused toner on the paper; thus, inkjet-printed characters have a wider black-to-white transition on their edges than laser-printed characters.

To illustrate the above differences, the word 'the' printed/copied by the three types of devices is shown in Figure 1. The differences in how the word is rendered can be observed when the scanned images are magnified. As the signal is transferred most accurately in the process of laser printing, the laser-printed word has a glossy appearance and a clear contour (Figure 1a). The quality of inkjet printing is lower than that of laser printing, as seen in the inkjet-printed word (Figure 1b). The copied characters (Figure 1c) possess a clearer contour than the inkjet-printed characters, although their quality is not as good as that of laser-printed characters because of the degradation caused by scanning a printed document.

In the next section, we develop an approach to differentiate between documents or characters produced by different devices based on the above observations. AWGN noise energy, impulsive noise energy, and contour roughness are used to differentiate between laser printers and electrostatic copiers. Contour roughness and average gradient along the contour are used to differentiate between laser-printer and inkjet-printer documents.

4 Proposed algorithm

In this section, we present the algorithm for distinguishing documents produced by laser printers, inkjet printers, and electrostatic copiers. We first give an overview of the different steps involved and then provide more details in the subsections that follow.

1. Pre-processing: As the proposed method is based on features derived from individual characters, the
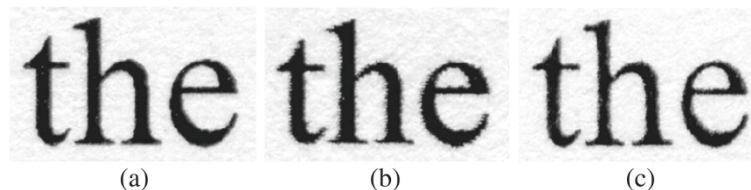
Figure 1 Representative word printed by each device. **(a)** Laser-printed, **(b)** inkjet-printed, and **(c)** electrostatic-copied.

characters in the tilt-corrected scanned image are first segmented. A threshold Th is generated to binarize the character image and used to simultaneously divide each character image into three parts: the text region, edge region, and background region.

2. Feature extraction: As the background region contains no information about the character, it is not considered in this paper. Instead, the following four features are extracted from the text and edge regions:

  a. Noise energy in the text region
  b. Noise energy in the edge region
  c. Contour roughness on the character
  d. Average gradient on the edge region of the character.

3. Classification and decision: After the four features are extracted from the character, SVM is applied to classify each of the characters. The origin of the document or a part of the document is then decided upon based on the individual character decisions using majority voting.

4.1 Image pre-processing

Before feature extraction and analysis, a scanned document image should undergo some pre-processing operations. First, each character of document image is segmented by tilt correction and character segmentation. A threshold Th is then applied to obtain a binary image; Th is computed by averaging the pixel values corresponding to the two peaks on the character histogram. For example, the binary image of Figure 1b is shown in Figure 2a. This binary image is used to extract contour roughness features.

Using the same threshold Th generated from the character histogram, the text, edge, and background regions are identified to extract the noise energy and average gradient features. In the experiments, pixel values in the range of $0.8Th\sim1.2Th$ were found to describe the edge region well for the three types of characters; thus, the edge region

Figure 2 Result of binarization and region classification. (a) The binary image of the word 'the'. **(b)** Text region (green), edge region (red), and background region (blue).

of characters is determined in this threshold range. Pixel values less than $0.8Th$ are classified as text region, while values greater than $1.2Th$ are classified as background region. Figure 2b shows the three regions obtained from Figure 1b in different colors: the text region is shown in green, the background region is shown in blue, and the edge region where pixels transfer from white to black is shown in red.

In order to remove the influence of the background noise, we discard the background region when extracting features. The noise energy features are computed from the text and edge regions of the character. Contour roughness is computed from the binary image, and average gradient is computed from the edge region.

4.2 Feature extraction

Due to the differences in the technical processes of printing/copying techniques and the resulting diversity in morphology, we compute discriminative features for these types of devices based on character morphologies including noise energy, contour roughness, and average gradient.

We selected features based on the following requirements:

  1. Independent of specific characters: the features can be extracted from arbitrary letters and words, and their values remain stable for different content;
  2. Independent of brand or model for printer and copier: for each type of technique, different brands or models may produce different character morphologies. The features are not expected to sensitive to the individual brands or models. Perfect features fall consistently in the feature space for each device type.
  3. Sensitive to printing and copying technique.

4.2.1 Noise energy in characters

As described in Section 3.1, laser printers and electrostatic copiers have similar architectures. Nevertheless, they produce significantly different printout qualities. Noise is introduced to the document during the scanning step in the copying process. This noise consists of AWGN and impulsive noise. AWGN noise mainly exists in the edge region of the character and has a strong influence on character morphology. In the edge region, a copied character has more significant texture than a printed character. The impulsive noise occurs mainly in the text region, appearing as white dots on the character. As described above, we extract the noise energy of AWGN and impulsive noise from the edge and text regions of the character, respectively. The noise energy in a copied character is expected to be greater than in a printed character.
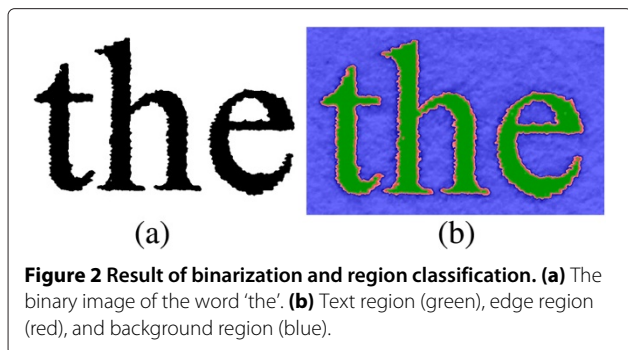
Discrete wavelet transform (DWT) is used to extract the noise described above. One-level DWT is performed on the character, and Gaussian and median filtering are conducted on the wavelet coefficients to remove noise in the edge and text regions, respectively. After filtering, the denoised image is obtained by re-constructing the wavelet coefficients. Finally, the Gaussian and impulsive noises are computed by subtracting the original image from the filtered image.

Denoising by applying a Gaussian filter to DWT coefficients Before denoising the edge region, the noise distribution and its parameters should be known. Assuming that the distribution of AWGN is $(0,\sigma)$ [22], the standard deviation σ can be estimated [23] as

$$\hat{\sigma} = \frac{\text{Median}\left(|W - \overline{W}|\right)}{0.6745}, \qquad (1)$$

where W denotes the wavelet sub-band coefficients in HH for the first level and denotes the average of W. As AWGN is present mainly in high-frequency coefficients, the standard deviation estimation is conducted on the wavelet sub-band coefficients HH in the first level.

An optimization algorithm is applied to estimate the AWGN in the image. The algorithm essentially performs repeated iterative calculations using a Gaussian filter to remove the AWGN in the image using filter parameters determined in the previous iteration. The initial standard deviation value is set to 0.5, and the iterative calculation is terminated when the difference between two successive estimates of standard deviation is less than a threshold ε. Because Σ is monotonically decreasing, the process converges.

The pseudo-code for the optimization algorithm is shown below. We start with iteration $k = 1$, where $\Sigma^k = 0.5$ and $\alpha = 0.5$. We use a convergence threshold of $\varepsilon = 0.001$.

Step 1: $I_{DN}^k = \text{IDWT}\left(\text{GF}\left(\text{DWT}\left(I_{ER}\right), \Sigma^k\right)\right);$

Step 2: $I_{AWGN}^k = I_{ER} - I_{DN}^k;$

Step 3: $\sigma_{AWGN}^k = \text{std}\left(I_{AWGN}^k\right);$

Step 4: $\Sigma^{k+1} = \Sigma^k + \alpha \cdot \frac{\sigma_{AWGN}^k - \hat{\sigma}}{\hat{\sigma}} \cdot \Sigma^k;$

Step 5: If $|\Sigma^{k+1} - \Sigma^k| < \varepsilon$
  compute I_{AWGN}^{k+1} and return
  else
  $k = k + 1$, go to **Step 1**,

where GF denotes the Gaussian filter function with parameter Σ and IDWT is the inverse DWT. The

Gaussian filter is applied on the horizontal component HH, the vertical component HV, and the diagonal component HD of the first-level DWT; the sizes of these filters are 1×9, 9×1, and 9×9, respectively. I_{ER} denotes the character image in the edge region, and I_{AWGN}^k and I_{DN}^k are the AWGN image and denoised image of I_{ER} in the kth iterative calculation, respectively.

Denoising by median filter on DWT coefficients During the experiment, some impulsive noise could be found in the smooth region of characters; it may due to the quality defects in a device. Most of this impulsive noise is so small that it could be seen only by magnifying the scanned image. Generally speaking, the location, quantity, and intensity of impulsive noise are stochastic. The quantity and intensity of impulsive noise are greater in a copied document compared to a printed document.

A median filter is used to estimate the impulsive noise in document text. As the text region is smooth, we apply a median filter on DWT coefficients in the text region of characters. The median filter is performed on the horizontal component HH, the vertical component HV, and the diagonal component HD of the first-level wavelet transformation using filter sizes of 1×9, 9×1, and 9×9, respectively. The denoised image of the text region is described by

$$I_{DN} = \text{IDWT}(\text{MF}(\text{DWT}(I_{SR}))), \qquad (2)$$

where I_{SR} denotes the text region of the characters and MF denotes the median filter function. The image of impulsive noise is given by

$$I_{IN} = I_{SR} - I_{DN} \qquad (3)$$

Noise energy A large amount of noise will be introduced into documents in the copying process, and the amount of noise in a copied document should thus be greater than in a printed document. In this paper, SVD is applied to describe the noise energy. Specifically, as the AWGN and impulsive noise are extracted from the edge and text regions of the character, respectively, we implement SVD and analyze the noise energy in these two regions. The SVD for matrix I is given by

$$I = U\Sigma V^T, \qquad (4)$$

where $U \in R^{m \times m}$ and $V \in R^{n \times n}$ are orthogonal matrices, $\Sigma \in R^{m \times n}$ is a diagonal matrix whose diagonal elements $\lambda^1 \geq \lambda^2 \geq \cdots \lambda^k \geq 0$ denote the eigenvalues of matrix I with $k = \min(m, n)$. The eigenvalues represent the energy that I projects to a subspace in U and R, and the eigenvalues in Σ are sorted by value from large to small. That

is to say, the energy projected to the subspace changes in descending order.

As the contents of different characters may affect the noise energy, we should remove the influence of character content from the noise energy before using it to identify printing and copying techniques. Thus, SVD is conducted on the noise image and denoised character image. Letting $\lambda_N^1, \lambda_N^2, \cdots, \lambda_N^k$, denote the eigenvalues of the noise image of the characters and $\lambda_{DN}^1, \lambda_{DN}^2, \cdots, \lambda_{DN}^k$, denote the eigenvalues of denoised characters, the summation of the ratio between noise and denoised character energies in different subspaces can be denoted as

$$E = \sum_{i=1}^{k} \frac{\lambda_N^i}{\lambda_{DN}^i} \qquad (5)$$

As AWGN and impulsive noise are extracted on the edge and text regions of characters, respectively, we compute the ratio of energy between the two regions and denote them as E_1 and E_2, respectively. These are the first two-dimensional features considered to distinguish printing and copying techniques.

4.2.2 Contour roughness

Roughness is used to distinguish inkjet printing from laser printing and copying. In inkjet printing, tails or satellites trail the ink drop in the scan direction because the drops of ink are fired onto paper when the print head is moving [5]. Due to these periodic tails or satellites, the character contour is rougher than in laser-printed and copied characters, and the value of contour roughness is higher for inkjet-printed characters. Roughness is computed based on the digital image of characters.

Coordinate extraction of character contour The edge searching method based on the binary image is applied to extract the character contour, and the coordinates of contour pixels are recorded in this process. The selection of the initial contour point is arbitrary; in this paper, we search rows from top to bottom and columns from left to right to find the initial point. The search terminates when the current searched pixel returns to the initial point.

Since each pixel has eight adjacent pixels, we search the adjacent pixels one by one to insure the searched pixels are on the contour of characters; the pixel then moves to the next adjacent one to search the next contour pixel. During the searching process, we record the coordinates of the contour pixels that are used to compute contour roughness. It should be noted that one character may have more than one connected region; we search the connected regions one by one and record the contour pixel coordinates until all the contour pixels of the character are recorded.

Character contour projection A projection is applied on the character contour to transfer the 2-D contour curve to a 1-D vector used to compute the contour roughness. Initially, each character contour is divided into several segments. The contour in each segment fragment could be seen as a 'straight' curve. Next, a line is generated that connects the two endpoints of each segment, and the distances between segment pixels and the line are computed. Finally, the distances for all the contour pixels are assigned to a 1-D vector. This process can be denoted as the character contour projection from 2-D contour to 1-D vector.

There exist some corner points and radians in each character, and the length of the contour segments should thus be restricted. A perfect contour segment contains no corner point and has a radian small enough to prevent deviation in the distance computation. The following iterative algorithm determines the length of the contour fragment.

Initial conditions: Suppose that the initial searching step L, which is the search length on the contour, is 1/5 of the character height. The initial searching point O is arbitrary on the contour. The starting and ending pixels of the current segment are A and B, respectively, where $|\overrightarrow{AB}| = L$. C denotes the arbitrary point on the contour between A and B, as illustrated in Figure 3. The selection of L and the distance computation are conducted as follows:

Step 1: *Compute the distance d from point C to line AB. If* $\max(d) > 5$, *go to* **Step 4**;

Step 2: *Compute the angle α when AC rotate clockwise to AB. If ∃C makes α > 180°, go to* **Step 4**;

Step 3: *Compute the angle β between $\overrightarrow{AC}$ and $\overrightarrow{BC}$. If ∃β > 0, go to* **Step 5**;

Step 4: $L = L - 1$;
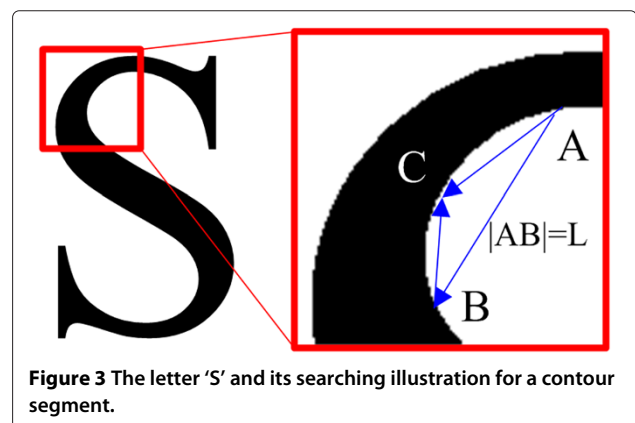
Figure 3 The letter 'S' and its searching illustration for a contour segment.

Step 5: *Compute the distance between the contour pixels on the segment and line AB, assign the distance values to a vector. If B is the initial searching point O, go to* **Step 7**;

Step 6: *Search the next end pixel of the segment using the step length of L. The new end pixel is defined as B and the original B is redefined as A. If the initial search point O is between A and B, define O as B and go to* **Step 1**;

Step 7: *Return the 1-D vector; end of the algorithm.*

In the iterative algorithm, steps 1 to 3 serve to constrain the length of the contour segment. These limitations ensure that the curvature on the contour segment is not large and that there is no corner on the contour segment. Curvature and corners have a significant influence on the contour roughness computation. Finally, a vector is obtained from the iterative algorithm.

Contour roughness The contour roughness of a character is defined as the mean value obtained from subtracting the maximum and minimum pixel values on a contour segment of fixed length. The size of the section is chosen as $m = 20$, which is greater than or equal to the period of the ink tails on the character contours. If n is the number of contour pixels, the contour curve can be divided into $\lceil n/m \rceil$ segments. Where P_i denotes the pixels in the ith pixel segment on the curve and $i = 1, 2, 3, \cdots, \lceil n/m \rceil$, the character contour roughness R is given by

$$R = \frac{1}{\lceil n/m \rceil} \sum_{i=1}^{\lceil n/m \rceil} (\max(P_i) - \min(P_i)) \qquad (6)$$

The character contour roughness of a laser printer is expected to be less than that of an inkjet printer because character contours originated from laser printers are smoother. This feature describes the degree of character contour roughness between the two printing techniques.

4.2.3 Average gradient on a character edge

The diffusion velocity of carbon is considerably different between print techniques, leading to significant differences in the gradient of pixel values in the edge regions of characters. The solid toner in a laser printer is melted and pressed on to the paper by the fuser; the toner diffuses slowly into the paper and solidifies quickly as the paper passes through the fuser. In an inkjet printer, the liquid ink is fired onto the paper and diffuses rapidly until it infiltrates completely into the paper. Thus, the black-to-white transition at the character edge is wider in inkjet printers than in laser printers. We use the average gradient of character edge to describe the black-to-white

transition velocity. Supposing that the character image is I, the gradient for the character image is denoted as

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \qquad (7)$$

The threshold Th for image digitization is computed during the pre-processing stage. The pixel value range of the edge region is defined as $0.8Th \sim 1.2Th$, and the pixels in the edge region constitute a collection M. The average gradient of the character edge G_{AVERAGE} in the edge region is given by

$$G_{\text{AVERAGE}} = \frac{1}{|M|} \sum_{(i,j) \in M} | \nabla I|, \qquad (8)$$

where $|M|$ denotes the number of elements in M and $| \nabla I|$ is the magnitude of ∇I.

Figure 4 shows the edge regions (blue color) of the letter 'e' printed by laser and inkjet printers with the same pixel value range. The region in Figure 4b is wider than that in Figure 4a, indicating that the black-to-white transition velocity in inkjet-printed characters is slower than that in laser-printed characters.

5 Experiments and results

The computer used in the experiment is a 3.40-GHz CPU with an Intel® Core™ i7-3770 processor and 16 GB of installed memory (RAM). Table 1 lists the printers and copiers used in our experiments. We chose ten laser printers, six inkjet printers, and nine electrostatic copiers to test algorithm performance. Ten pages consisting of English words and letters were printed or copied for each device in the experiments. The documents were scanned with resolutions ranging from 300 to 1,200 dpi, which is the most commonly used range of resolutions in practice. The flatbed scanner used in the scanning work was an Epson Perfection V33 (Seiko Epson Corp., Suwa, Nagano, Japan). Scanned images were saved as gray
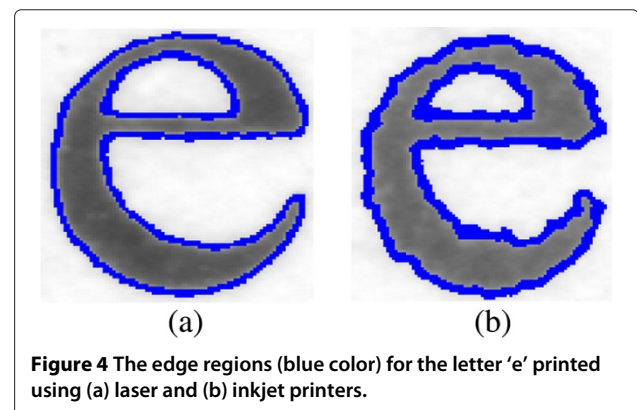
Figure 4 The edge regions (blue color) for the letter 'e' printed using (a) laser and (b) inkjet printers.

Table 1 Printers and copiers used in the experiments

Laser printer		Inkjet printer		Copier	
Label	Brand and model	Label	Brand and model	Label	Brand and model
L1	Canon LBP3500	I1	Epson T50	C1	Canon IR6000
L2	HP CP1525n	I2	Epson R280	C2	Imagistics im9220
L3	HP CP3800dtn	I3	Epson R290	C3	Kyocera km 8030
L4	HP 4240n	I4	Epson Tx650	C4	Lanier LD360
L5	Lanier LP540C	I5	Epson 1390	C5	Lanier LD528
L6	Lenovo LJ6000	I6	Canon iX4000	C6	Oce 9220
L7	Samsung 3471nd			C7	Toshiba 4540C
L8	HP Pro 400 MFP			C8	Konica Minolta 7272
L9	HP 1505n			C9	Konica 7075
L10	HP CP3525dn				

Canon Inc., Ohta-ku, Tokyo, Japan (Canon and Oce); Seiko Epson Corp., Suwa, Nagano, Japan; Hewlett-Packard Company, Palo Alto, CA, USA; Imagistics International Inc., Denver, CO, USA; Konica Minolta Inc., Marunouchi, Chiyoda, Tokyo, Japan; Kyocera Corp., Kyoto, Japan; Ricoh Americas Corp., Malvern, PA, USA (Lanier); Lenovo Group Ltd., Beijing, China; Samsung Group, Samsung Town, Seoul, South Korea; Toshiba Corp., Tokyo, Japan.

images in TIFF and JPEG formats with 8-bit depths. We used the LIBSVM package [24], and a radial basis function (RBF) was chosen as the kernel function. SVM training and testing were repeated ten times, and the results were averaged. The parameters from Section 4, including the threshold in pre-processing, filter sizes, and the parameters in optimization method, are constant in the following experiments.

5.1 Performance of the features

We now test the performance of the features extracted using the techniques and parameters described in Section 4. First, 500 characters were randomly selected

from the documents produced by each printer and copier in our dataset. Figure 5 shows the scatter plot of noise energy and contour roughness for the laser printers and copiers. The laser-printed characters are labeled in blue and denoted by 'L#', while the copied characters are labeled in red and denoted by 'C#'. Their difference is not readily apparent in the plot because of plot overlap, and it is difficult to distinguish them by linear classification.

Figure 6 shows the scatter plot of contour roughness and average gradient for laser (blue, 'L#') and inkjet (red, 'I#') printers. The scatter plot for the inkjet printer is easily differentiated due to its centralization. The feature values in Figures 5 and 6 are normalized.
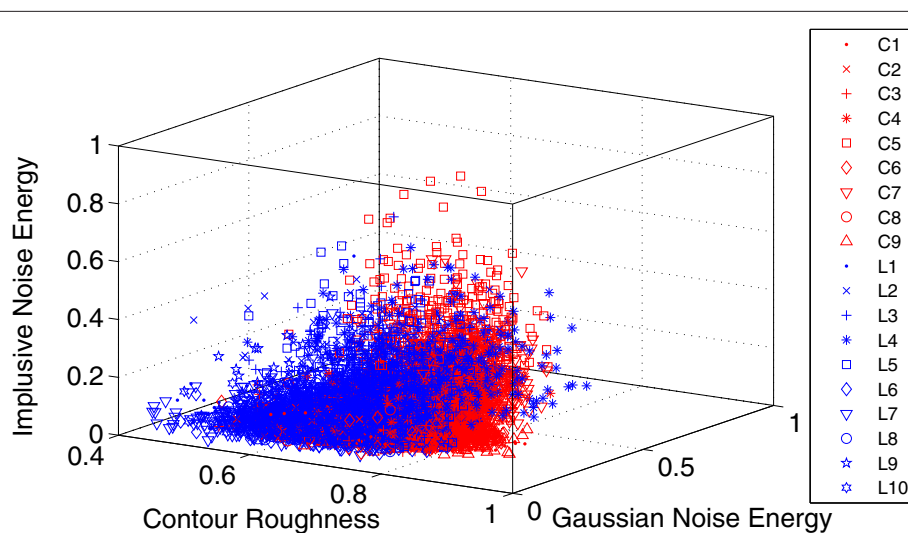
Figure 5 Scatter plot of contour roughness and average gradient for laser printers and copiers.
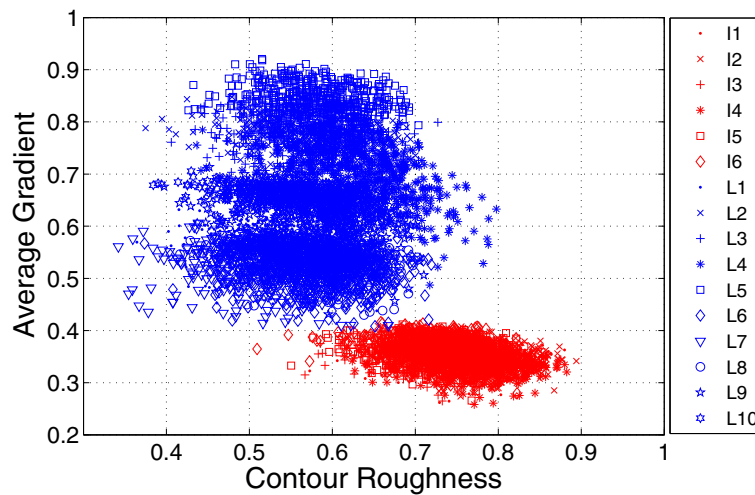
Figure 6 Scatter plot of contour roughness and average gradient for laser and inkjet printers.

5.2 Experimental results

A set of experiments was conducted to test the performance of our algorithm. Scanned TIFF documents with resolutions of 1,200 dpi were used for these tests. We printed or copied ten pages of English words and letters for each device listed in Table 1. Tables 2, 3 and 4 show the document source type identification accuracy for laser printers, inkjet printers, and electrostatic copiers, respectively. For the two iterative algorithms used in this method, it needed about 1.5 s to identify each character or word. For the whole page detection, the detection time depended on the number of words in the document.

We chose 540, 900, and 600 words and letters for each printer and copier to train the model; the total number for each type of device was 5,400. The words and letters used for training were chosen randomly from the database, and the remaining words and letters were used for testing. The experiments were repeated ten times, and Tables 2, 3 and 4 report the average accuracies. The first columns of the tables show the actual labels for the printers and copiers; the first rows indicate the computed decisions for the three types of devices by the proposed algorithm.

We also investigated the stability of the trained SVM models. Figure 7 shows the source device type identification results for each device. The blue bars denote results from the models trained by all the devices where the results for ten tests were averaged. The minimum and maximum prediction results are labeled in the form of error bars on the top of the blue bars. The source type identification accuracy for inkjet printers is the highest due to their signatures being very distinct from the others. Since the signatures for the laser printer and electrostatic copier are quite similar, the accuracies for 'L4,' 'L6,' and 'C3' are very low. This result may also be due to the toner quality.

Table 2 The classification accuracy for laser printers (%)

		Predicted		
		Laser printer	Inkjet printer	Copier
	L1	92.31	0.05	7.64
	L2	94.98	0	5.02
	L3	94.27	0	5.73
	L4	70.89	0	29.11
Actual	L5	91.75	0	8.25
	L6	65.71	0.11	34.18
	L7	93.04	0.05	6.91
	L8	99.00	0.03	0.97
	L9	99.82	0	0.18
	L10	99.99	0	0.01

Table 3 The classification accuracy for inkjet printers (%)

		Predicted		
		Laser printer	Inkjet printer	Copier
	I1	0.01	99.50	0.49
	I2	0.01	98.31	1.68
Actual	I3	0.04	97.13	2.83
	I4	0	99.87	0.13
	I5	0.04	92.31	7.65
	I6	0.03	99.19	0.78

Table 4 The classification accuracy for electrostatic copiers (%)

		Predicted		
		Laser printer	Inkjet printer	Copier
	C1	0.99	12.13	86.88
	C2	4.10	3.54	92.36
	C3	39.89	0.15	59.96
	C4	18.28	0	81.72
Actual	C5	12.52	0	87.48
	C6	0.74	4.75	94.51
	C7	5.27	0	94.73
	C8	6.07	1.72	92.21
	C9	3.89	2.29	93.82

Next, to test the accuracy of the algorithm on unknown devices, we established a back-to-back model. Taking 'L1' as an example, we chose the training features from all the devices except 'L1' and used this classification model to detect 'L1'. The resulting accuracies are shown as green bars in Figure 7. For some devices, the detection accuracy is sensitive to the model (e.g., 'L4', 'C1', 'C3', 'C4', and 'C7'); the accuracies for these devices drop dramatically compared to the average accuracies. However, for 20 of the devices, the accuracy is unaffected by the model.

5.3 The effect of JPEG compression
In order to examine the robustness of our algorithm, experiments were conducted in which the scanned documents were compressed into JPEGs at different compression qualities. The scanned images (1,200 dpi) were saved in JPEG format with the compression qualities ranging from 30 to 100. The bar graph in Figure 8 shows the comparative performance of different JPEG compression rates and devices, demonstrating that the proposed features are robust to JPEG compression.

5.4 The influence of scanning resolution
In practice, the document to be detected can have low resolution; therefore, sensitivity to resolution is another important measure for the performance of any forensics technique. We tested the robustness of the proposed algorithm using resolutions ranging from 300 to 1,200 dpi, which covers the commonly used resolutions. The resulting device type identification performance is shown in Figure 9. The accuracy for inkjet printer detection is stable when the resolution is greater than 500 dpi. The accuracy for laser printer and copier detection grows steadily when resolution increases above 300 dpi. Although the SVM models are stable for 1,200 dpi resolutions, the model stability performs poorly when resolution is low (300 to 500 dpi). Thus, we conclude that the proposed algorithm is not sufficiently discriminative to identify source device types at lower resolutions.

5.5 Forgery detection
As previously mentioned, an algorithm that classifies source device type based on features of individual letters or words can also be used for forgery detection where a forged document contains characters originated from different device types. To test this, we created a forged document and executed the proposed algorithm. Figure 10 shows a tampered document and its detection result. The original document is a contract written English and printed by a laser printer. Figure 10a shows a
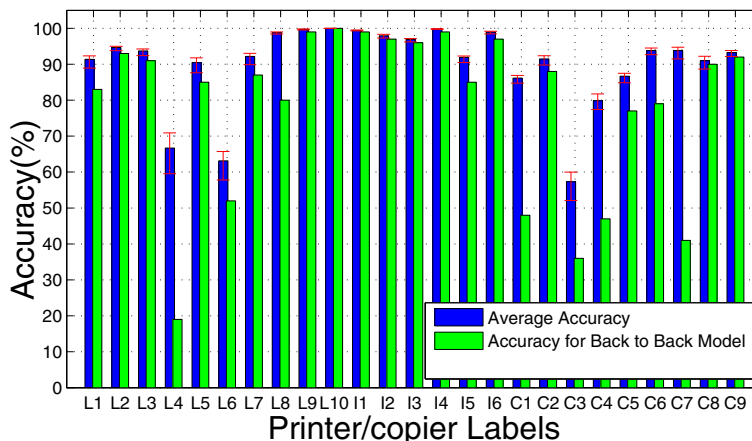
Figure 7 The accuracy of device identification.
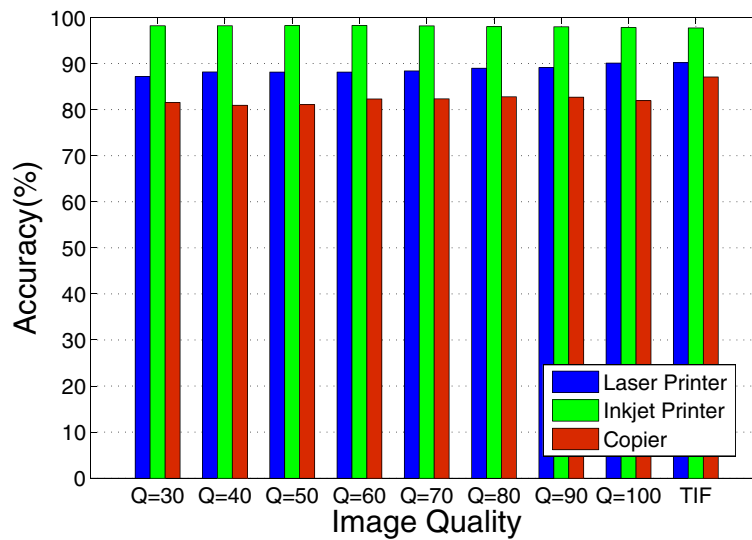
**Figure 8 Comparative performances for different JPEG compression qualities.**

one-page tampered document in which the item '6. Insurance: To be covered by the Seller.' was added at the bottom of the contract. The additional item may be add by printing with an inkjet printer or by splicing two parts together followed by copying. The predicted laser-printed words are labeled by red squares, the predicted inkjet-printed words are labeled by green squares, and the predicted copied words are labeled by blue squares. Figure 10b shows the detection result of Figure 10a tampered by adding inkjet printing; only the tampered region is labeled by green squares. Figure 10c shows the detection result of Figure 10a tampered by splicing and copying; most of the

words are labeled by blue squares, and only a few words are classified incorrectly. In addition, the two signatures are ignored when segmenting the characters.

6 Conclusions

In this paper, we have proposed a method for detecting fraudulent documents. Characters in the document are analyzed to identify the signatures of the common device(s) used to create them. These devices could include laser printers, inkjet printers, and electrostatic copiers.

As the devices differ in their technical architectures, they produce characteristic character morphologies in the
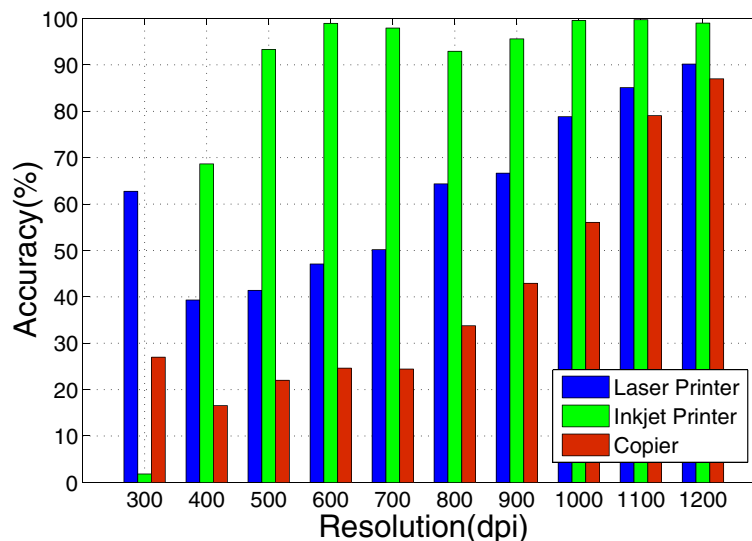
Figure 9 Algorithm performance at different resolutions.

Figure 10 Tampered document (a contract) and its detection results. (a) The document tampered by add-printing or splice-copying, **(b)** the detection result for add-printing, and **(c)** the detection result for splice copying.

document. To identify the source-specific characteristics, we extract features including AWGN energy, impulsive noise energy, contour roughness, and average gradient from the characters. The AWGN and impulsive noise energies are isolated by wavelet transformation (WT) and SVD on the character image and used to distinguish between laser printers and copiers. Contour roughness is drawn from the character contour; it reflects the degree of roughness on the character contour and is mainly used for laser and inkjet printer identification. Average gradient is used to distinguish inkjet printers from laser printers as the black-to-white transition velocity is greater in inkjet-printed characters.

The experimental results show the effectiveness of the proposed method. For all the inkjet printers and most laser printers and copiers, the accuracy reaches 90%. The SVM models were tested for stability, and the method is also robust to JPEG compression. However, the method is sensitive to resolution, with better accuracy obtained at higher resolutions. Finally, since the proposed technique utilizes the classification of individual letters or words, it was shown that it can be used to detect a forged document consisting of parts originating from different device types.

Future work will focus on improving the robustness of the method by making it work with all brands of devices and improving its accuracy at lower resolution.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

SS contributed to the study design, study concepts, literature research, data acquisition, data analysis/interpretation, experimental studies, manuscript preparation, manuscript editing, manuscript revision/review, and manuscript final version approval. NM contributed to the study design, manuscript editing, manuscript revision/review, and manuscript final version approval. XK contributed to the study design, study concepts, literature research, manuscript preparation, and manuscript final version approval. All authors read and approved the final manuscript.

Author details

[1] Dalian University of Technology, No. 2 Linggong Road, Ganjingzi District, Dalian 116024, China. [2] Polytechnic School of Engineering, New York University, 6 MetroTech Center, New York City, NY 11201, USA.

References

1. HT Sencar, N Memon, *Digital Image Forensics: There Is More to a Picture than Meets the Eye*. (Springer, New York, 2013), pp. 1–370
2. AK Mikkilineni, N Khanna, EJ Delp, Forensic printer detection using intrinsic signatures, in *Proceedings of SPIE-IS and T Electronic Imaging - Media Watermarking, Security, and Forensics III* (San Francisco, CA, USA, 24–26 Jan 2011), pp. 78800–11
3. AK Mikkilineni, PJ Chiang, G Ali, GTC Chiu, P Allebach, EJ Delp, Printer identification based on graylevel co-occurrence feature for security and forensics application, in *Proceedings of the SPIE - The International Society for Optical Engineering* (San Jose, CA, USA, 21 Mar 2005), pp. 430–440
4. AK Mikkilineni, N Khanna, EJ Delp, Texture based attacks on intrinsic signature based printer identification, in *Proceedings of the SPIE - The International Society for Optical Engineering* (San Jose, CA, USA, 18–20 Jan 2010), pp. 75410–12
5. PJ Chiang, N Khanna, AK Mikkilineni, MVO Segovia, S Suh, JP Allebach, GTC Chiu, JE Delp, Printer and scanner forensics. IEEE Signal Process. Mag. **26**, 72–83 (2009)
6. GN Ali, AK Mikkilineni, JP Chiang, GT Allebach, EJ Delp, Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices, in *International Conference on Digital Printing Technologies* (New Orleans, LA, USA, 28 Sept–3 Oct 2003), pp. 511–515
7. YB Wu, XW Kong, YP Guo, Printer forensics based on page document's geometric distortion, in *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP 2009)* (Cairo, Egypt, 7–12 Nov 2010), pp. 2909–2912
8. O Bulan, J Mao, G Sharma, Geometric distortion signatures for printer identification, in *ICASSP 2009 - 2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (Taipei, Taiwan, 19–24 Apr 2009), pp. 1401–1404
9. L Cui, Document inspection forged by photocopying. J. Chinese People's Public Secur. Univ. (Science and Technology). **3**, 22–24 (2008)
10. JV Beusekom, F Shafait, TM Breuel, Document inspection using text-line alignment, in *ACM International Conference Proceeding Series* (Boston, MA, USA, 9–11 Jun 2010), pp. 263–270

11. JV Beusekom, F Shafait, Distortion measurement for sutomatic document verification, in *Proceedings of the 2011 11th International Conference on Document Analysis and Recognition (ICDAR 2011)* (Bejing, China, 18–21 Sept 2011), pp. 289–293

12. E Kee, H Farid, Printer profiling for forensics and ballistics, in *MM and Sec'08: Proceedings of the 10th ACM Workshop on Multimedia and Security* (Oxford, UK, 22–23 Sept 2009), pp. 3–9

13. JS Aronoff, SJ Simske, Effect of scanner resolution and character selection on source printer identification. J. Inform. Sci. Technol. **55**, 506021–5060210 (2011)

14. H Gou, A Swaminathan, M Wu, Robust scanner identification based on noise features, in *Proceedings of SPIE - The International Society for Optical Engineering* (San Jose, CA, USA, 29 Jan–1 Feb 2007), pp. 0–0

15. N Khanna, AK Mikkilineni, EJ Delp, Scanner identification using feature based processing and analysis. IEEE Tran. Inform. Forensics Secur. **4**, 123–139 (2009)

16. AE Dirik, HT Sencar, N Memon, Flatbed scanner identification based on dust and scratches over scanner platen, in *ICASSP 2009 - 2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (Taipei, Taiwan, 19–24 Apr 2009), pp. 1385–1388

17. J Tchan, Classifying digital prints according to their production process using image analysis and artificial neural networks, in *Proceedings of the SPIE - The International Society for Optical Engineering* (San Jose, USA, 27–28 Jan 2000), pp. 105–116

18. J Tchan, The development of an image analysis system that can detect fraudulent alterations made to printed images, in *Proceedings of the SPIE - The International Society for Optical Engineering* (San Jose, USA, 20–22 Jan 2004), pp. 151–159

19. CH Lampert, TM Breuel, Printing technique classification for document counterfeit detection, in *2006 International Conference on Computational Intelligence and Security* (Guangzhou, China, 3–6 Nov 2006), pp. 639–644

20. M Umadevi, A Agarwal, R Rao, Printed text characterization for identifying print technology using expectation maximization algorithm. Multi-disciplinary Trends Artif. Intell. **7080**, 201–212 (2011)

21. C Schulze, M Schreyer, A Stahl, T Breuel, Using DCT features for printing technique and copy detection. Adv. Digital Forensics V. **306**, 95–106 (2009)

22. TV Hoang, H Elisa, B Smith, S Tabbone, Edge noise removal in bi-level graphical document images using sparse representation, in *18th IEEE International Conference on Image Processing (ICIP 2011)* (Brussels, Belgium, 11–14 Sept 2011), pp. 3549–3552

23. DL Donoho, IM Johnstone, Ideal spatial adaptation via wavelet shrinkage. Biometika. **81**, 425–455 (1994)

24. C Chang, C Lin, LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **27**, 1–27 (2011)