Texture feature benchmarking and evaluation for historical document image analysis

Maroua Mehri, Pierre Héroux, Petra Gomez-Krämer, Rémy Mullot

IJDAR manuscript No.
(will be inserted by the editor)

Texture Feature Benchmarking and Evaluation for Historical Document Image Analysis

Maroua Mehri · Pierre Héroux · Petra Gomez-Krämer · Rémy Mullot

Received: 13 February 2016 / Revised: 28 September 2016 / Accepted: 19 December 2016

Abstract The use of different texture-based methods is pervasive in different sub-fields and tasks of document image analysis and particularly in historical document image analysis. Nevertheless, faced with a large diversity of texture-based methods used for historical document image analysis, few questions arise. Which texture methods are firstly well suited for segmenting graphical contents from textual ones, discriminating various text fonts and scales, and separating different types of graphics? Then, which texture-based method represents a constructive compromise between the performance and the computational cost? Thus, in this article a benchmarking of the most classical and widely used texture-based feature sets has been conducted using a classical texture-based pixel-labeling scheme on a large corpus of historical documents to have satisfactory and clear answers to the above questions. We focus on determining the performance of each texture-based feature set according to the document content. The results reported in this study provide firstly a qualitative measure of which texture-based feature sets are the most appropriate, and secondly a useful benchmark in terms of performance and computational cost for current and future research efforts in historical document image analysis.

Maroua Mehri · Pierre Héroux
Normandie Univ, UNIROUEN, UNIHAVRE, INSA Rouen, LITIS, 76000 Rouen, France
Tel.: +33-2 32 95 50 11
Fax: +33-2 32 95 50 22
E-mail: maroua.mehri@gmail.com, pierre.heroux@univ-rouen.fr

Petra Gomez-Krämer · Rémy Mullot
L3i EA 2118, University of La Rochelle, Avenue Michel Crépeau, 17042, La Rochelle, France
Tel.: +33-5 46 45 82 62
Fax: +33-5 46 45 82 42
E-mail: {petra.gomez, remy.mullot}@univ-lr.fr

Keywords Benchmarking · Texture · Pixel-labeling · Historical document image analysis

1 Introduction

Since the early 1990s, libraries and museums have conducted large digitization campaigns with cultural heritage documents and scientific resources. The goals of these large digitization campaigns consist in ensuring restoration and lasting preservation of the cultural patrimony, and promoting its worldwide accessibility. The cultural patrimony requires to be protected from further deterioration and damages caused by repetitive handling [6]. Due to the huge amount of numeric high quality reproductions induced by the rapid growth of digital libraries worldwide, many challenges and open issues have been raised and have already spawned novel approaches and rigorous techniques of mass management. These solutions are designed to optimize the accessibility and navigability of huge mass and ever-increasing amount of available document images (DIs). Recently, raising interest to document image analysis (DIA) and historical DIA (HDIA) has been generated, since it helps to reach the objective of ensuring the indexing and retrieval of digitized resources and offering a structured access to large sets of cultural heritage documents [20]. DIA consists in dividing a DI layout according to the nature of the extracted structure such as separate text from non-text regions or partition text into columns, text blocks, lines, words, *etc.* It starts by segmenting a DI in order to find and classify homogeneous regions or zones, such as graphic and textual regions [60]. As stated by Kise [42], the analysis of pages with constrained layouts (e.g. rectangular, Manhattan) and clean DIs has almost been solved, while HDIA is still an open problem due to their particularities related to [20]:

- ***Properties***: Large variability of page layout, complicated and complex page layout (several columns with irregular sizes, dense printing, irregular spacing, marginal notes).
- ***Life cycle***: Noise and degradation caused by copying, scanning or aging (yellow pages, ink stains, mold or moisture, faded out ink, uneven lighting due to folded, corrugated parchment or papyrus), superimposition of information layers (stamps, handwritten notes at the margins, noise, back-to-front interference, ink that was bleeding through).
- ***Digitization***: Page skew, scanning defects (curvature, light, blur), presence of black borders.

It has recently been shown that texture analysis plays a fundamental role for HDIA since it has been considered as a consistent choice for meeting the need to segment a page layout under significant degradation levels and different noise types [52]. Kise [42] also precised that the most relevant methods used to analyze pages with unconstrained layouts and overlapping layers are based on signal properties of page components by investigating texture-based features and techniques. Hence, texture-based methods address the challenges of the existing state of the art and those initially dedicated to contemporary DIs. The use of texture analysis techniques for historical document images (HDIs) has become an appropriate choice, given that there are significant degradations and no hypothesis concerning the HDI layout and graphical properties or typographical parameters of the analyzed HDI, such as the type of script or handwriting (e.g. machine-print or printed, hand-print or manuscript, cursive), font size and type, scanning resolution, DI size, language, alphabet. For this reason, during the last two decades several texture-based feature sets have been investigated and shown to be robust when they have been extracted and analyzed from unconstrained and degraded DIs [40].

In spite of an invaluable number of different texture-based studies and contributions achieved on different subfields and tasks of pattern recognition, there is a very limited number of comparative studies of texture-based approaches in the fields of DIA and particularly HDIA. Those texture-based approaches have been reported as relevant and dedicated to a specific application and fine-tuned to a particular dataset. Nevertheless, the question of how these texture-based algorithms are compared with each other has not been properly addressed for HDIA. This is mostly due to the unavailability or lack of a standard public dataset of HDIs and its associated ground truth [3]. Thus, in this article we aim at providing a qualitative measure of which texture-based feature sets are the most appropriate both in terms of HDIA performance and computational cost. The four main contributions of this article are summarized in the following.

(1) A review of the different texture-based methods proposed in the state of the art, with a particular focus on those related to DIA and to HDIA.

(2) A description of the different texture-based feature sets evaluated in this article to highlight our choice of the values of the thresholds and the parameters set when extracting the assessed texture descriptors and after referring to the most common texture-based methods used with HDIs in the literature.

(3) The particularities and characteristics of a novel dataset of 1000 ground truthed one-page HDIs, which is publicly available for scientific use.

(4) A qualitative measure of which texture-based feature sets are the most appropriate after presenting a useful benchmark and informative comparative performance evaluation under realistic circumstances in terms of the pixel-labeling performance and the computational cost for current and future research efforts in HDIA.

In this article, parts of the work presented in [50,53] have been reported. The comparative study presented in [50, 53] has been extended by adding three more texture-based feature sets and explaining our evaluated texture features in more detail than in [50, 53]. Additionally, the experiments in this study have been carried out on two publicly available datasets (*i.e.* more than 3 times the size of the evaluated HDIs in [50,53]), illustrating with visual examples and demonstrating the performance of each texture-based feature set, along with the computational cost. We have also included several clustering and classification accuracy measures. Besides, a correlation analysis of the performance of each texture-based feature set has been presented to highlight the similarities of the behavior of the different evaluated texture features. Finally, a statistical comparison of the performances of the nine analyzed texture-based feature sets in this study has been proposed to validate the obtained results over two different datasets.

The remainder of this article is organized as follows. Section 2 reviews the different texture-based methods proposed in the literature, with a particular focus on those related to DIA and HDIA. Section 3 presents a brief description of the different texture-based feature sets evaluated in this article. In Section 4, we outline the experimental protocol by describing the corpus, the defined ground truth and the used pixel-labeling scheme for comparing the texture features. In section 5, qualitative results are firstly given to demonstrate the performance of each texture-based feature set, along with the computational cost (e.g. resources in terms of the memory requirements, the complexity and the time consumption considerations) to highlight the strengths and the weaknesses of nine well-known texture-based feature sets for HDIA. Then, we discuss quantitatively the obtained performance of the texture feature analysis experiments. Moreover, correlation and statistical analyses have been presented to examine the behavior of the different texture-based feature sets and validate their performance.

Finally, our conclusions and future work are presented in Section 6.

2 Texture analysis from image to historical document analysis: The state of the art

It is widely believed that analyzing texture features on images is relevant for many applications in image processing and pattern recognition fields [75]. Yet, texture has remained a relevant processing tool for the analysis of many types of images. Texture is considered by Haralick [32] as an important characteristic for the analysis of many kinds of images. Even if there is no precise definition of texture, many applications in various areas (e.g. biomedical image processing, industrial automation, remote sensing, DIA) have benefited of the texture-based algorithms proposed in the literature. Later, a general definition of texture was given as a measure of the variation in intensity, measuring properties such as smoothness, coarseness and regularity [37].

Classically, texture features are extracted and analyzed by using a texture-based method in order to generate a partition of the analyzed image into regions. The obtained regions have homogeneous characteristics and similar properties with respect to the extracted texture features [8]. Okun and Pietikäinen [60] assumed that text regions have different texture features from non-text ones. Indeed, text areas contain text lines sharing similar characteristics (e.g. approximately similar orientation, inter-character and interline spacings). This means that text regions are considered as regular and periodic textures while non-text ones are characterized by irregular textural properties. Thus, in our study one assumption is made to ensure a differentiation between different content types. Indeed, textual regions in a digitized DI are considered as textured areas, while its non-text content is considered as regions with different textures. Then, textual regions with different fonts are also distinguishable by means of texture analysis. Moreover, different types of graphics can be perceived as different textures (e.g. drop cap, embellishment, frame, illumination, engraving, stamp, sketch).

Okun and Pietikäinen [60] classified texture-based layout analysis approaches into two categories, "Group 1" and "Group 2". The first class of texture-based methods which is called "Group 1", is firstly processed by extracting document regions using smearing techniques. Then, each region is classified according to the extracted texture features. This category of methods has the disadvantage that its performance depends on the quality of the region extraction phase. The second class of methods which is called "Group 2", is processed by extracting texture features from a given analysis window. Since the "Group 2" of texture-based methods has been considered as a local processing technique, Okun and Pietikäinen [60] stipulated that this class of methods is more robust to different document layouts and/or DI skew than the "Group 1". They pointed out that the main issue of using texture-based methods is their quite high computational complexity. Especially, their processing time depends even more on the image size and resolution due to the use of pixel-based computation, large DI size and high complexity of texture analysis approaches. On the other side, Cote and Albu [19] classified the most widely used texture-based methods by the DIA community into two categories, the statistical and spectral methods. The statistical approaches investigate the spatial distribution of gray-levels within a region of interest, while the spectral ones describe texture by frequency descriptors obtained by computing the response of an image to a given filter bank. In our view, texture feature extraction and analysis methods which have been used on different sub-fields and tasks of DIA (e.g. pre-processing, character recognition, page decomposition), may be categorized into five classes according to the properties or characteristics of the extracted texture features [13]:

- ***Statistical feature-based methods***: They are used to analyze the spatial distribution of gray levels by computing local indices in the image and deriving a set of statistics from the distribution of the local features. The statistical methods have the advantage of being simple to implement and their effectiveness is shown. The auto-correlation function [32], GLCM [33], gray-level run-length matrix (GLRLM) [28] and Tamura [68] are few standard statistical methods.
- ***Geometric feature-based methods***: They are used to describe intricate patterns and to retrieve and describe texture primitives by characterizing the notion of a texton. Texture primitives may be extracted using a difference-of-Gaussian filter, for example [70]. Those methods attempt to characterize the primitives and find rules governing their spatial organization. Among the classic geometric methods, moment-based texture segmentation is one of the well-known methods [78].
- ***Model-based methods***: They are used to compute a parametric generative model based on the intensity distribution of texture primitives. A widely used class of the model-based methods are the probabilistic models. The conditional random fields (CRF) [45], LBP [59] are the most commonly used tools based on probabilistic models.
- ***Spectral feature-based methods***: They are used to investigate the overall frequency content of an analyzed image. The most widely used spectral methods in indexing and segmentation of natural images are Gabor filters [38] and wavelet transform [49].
- ***Hybrid feature methods***: They combine different kinds of texture features and other types of descriptors (e.g. shape, color, topological or spatial descriptors) to address a general issue in image segmentation and analysis [71].

The most common texture-based methods used with HDIs in the literature are summarized in Table 1.

Table 1: Texture-based methods used with HDIs in the literature.

Feature	Application
Tamura	– Zone classification [41] – Content segmentation [55]
LBP	– Text localization [9] – Pixel classification [15, 16] – Printed script identification [26] – Arabic font recognition [57]
GLRLM	– Pixel classification [36] – Zone classification [41] – Text line, word and character segmentation [58] – Handwritten annotation discrimination [66] – Content segmentation [73]
Auto-correlation	– Geometric layout analysis [18] – Handwriting classification [25] – Text recognition [29] – Layout analysis and content enrichment [31] – Pixel-labeling for historical books [40, 51]
GLCM	– Content segmentation and classification [48, 54] – Lettrine retrieval [71]
Gabor	– Detection of main text area from side-notes [5] – Pixel classification [16] – Text region segmentation [21] – Text/drawing separation [25] – Text detection [40]
Wavelet	– Text localization [44] – Content classification [43]

Due to the large diversity of texture-based methods, we conclude that there is a critical need to explore and compare various aspects of texture features by using a classical texture-based pixel-labeling scheme in order to assist HDIA and clarify a number of issues. Which texture features are firstly well suited for segmenting graphical contents from textual ones, discriminating text in a variety of situations of different fonts and scales and separating different types of graphics? Then, which texture features represent a constructive compromise between the performance (*i.e.* pixel-labeling quality) and the computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality)? Indeed, the performance of a texture-based segmentation method tightly depends on the type of the applied texture features.

Numerous surveys and comparisons of texture-based techniques have been proposed for image segmentation and analysis in the literature a few years ago. For example, Weszka *et al.* [76] compared different texture analysis methods based on the Fourier power spectrum, second-order gray-level statistics and first-order statistics of gray-level differences for terrain classification. They concluded that the first and second order statistics perform significantly better than the spectral approaches. A well-researched survey and complete overview of recent texture segmentation and feature extraction techniques for unsupervised applications was presented in [75], including Gabor filters, GLCM, fractals. They concluded that texture-based methods have distinct applications. Indeed, some model-based texture methods are suitable for stochastic textures, while some spectral-based texture methods (e.g. Gabor filters) are adequate for stochastic and structural textures. However, they did not present a quantitative comparison of the surveyed texture-based methods since they stated that is a demanding and time-consuming task. Few limited studies attempted to present quantitative comparisons of texture-based algorithms [24]. Okun and Pietikäinen [60] reported mainly visual or qualitative results of seven texture-based methods (run-lengths, multi-channel Gabor filters, texture co-occurrence spectrum, white tiles, texture masks, structured wavelet packet analysis and laws masks) to review the progress achieved for DI layout analysis. The reviewed methods were evaluated on magazines and newspapers (gray-scale or binary images). A comparative study has been carried out for selecting the texture feature category based on the best trade-off between the best performance and the lowest computation time [53]. This comparative study presented six texture-based feature sets (auto-correlation function, GLCM, Gabor filters, 3-level Haar wavelet transform, 3-level wavelet transform using 3-tap Daubechies filter and 3-level wavelet transform using 4-tap Daubechies filter) which have been assessed on only 314 historical documents.

Therefore, after referring to the most common texture-based methods used with HDIs in the literature (*cf.* Table 1) and based on the proposed categorization of the texture feature extraction and analysis methods, nine well-known and widely-used texture-based feature algorithms for HDIA have been selected and investigated in this article by detailing qualitative and numerical experiments on 1000 historical document images from the French digital library Gallica[1] and 100 pages used in the historical book recognition competition (HBR)[2]. In this study, we have chosen to investigate and compare basically statistical, spectral and model-based methods, is justified by the following reasons. Firstly, the investigated texture-based feature sets can be analyzed without using a learning phase. Moreover, the extraction of these texture features needs less parameter settings. Indeed, without hypothesis on either the DI layout or content, the choice of numerous appropriate thresholds and parameters is a very difficult task. In addition, the pre-defined parameters used when extracting and analyzing the texture features in our study are set up based on work published in the literature (Tamura [41, 55], local binary patterns [9], gray-level run-length matrix [73], auto-correlation function [40, 51], gray-level co-occurrence matrix [11], Gabor filters [38], 3-level Haar wavelet transform, 3-level wavelet transform using 3-tap Daubechies filter and 3-level wavelet transform using 4-tap Daubechies filter [44]). In addition, they are the most classic and common ones in the literature. Finally, they are well suited to any type of DIs and they have been widely

[1] `http://gallica.bnf.fr`

[2] `http://www.primaresearch.org/datasets`

investigated for a long time in independent experiments in order to segment and characterize DIs or part of them (*cf.* Table 1).

3 Texture features

The texture-based feature sets which are compared and evaluated in this study have been derived from the Tamura, local binary patterns (LBP), gray-level run-length matrix (GLRLM), auto-correlation, gray-level co-occurrence matrix (GLCM), Gabor filters and three wavelet-based approaches: 3-level Haar wavelet transform (Haar), 3-level wavelet transform using 3-tap Daubechies filter (Db3) and 3-level wavelet transform using 4-tap Daubechies filter (Db4). The remainder of this section summarizes the nine sets of texture features.

3.1 Tamura features

Tamura *et al.* [68] proposed to extract texture features corresponding to the human visual perception. They defined six basic texture descriptors, namely coarseness, contrast, directionality, line-likeness, regularity and roughness. They showed that the three first texture features (*i.e.* coarseness, contrast and directionality) consistently outperformed others for global descriptions of textures both separately and in different combinations for image segmentation and classification issues.

Four Tamura descriptors are extracted in this study, namely:

- Coarseness (I_t^1, *cf.* equation (9)): This feature characterizes the texture scale and repetition rates. Specifically, it measures the largest size at which a texture exists. It is considered by Tamura *et al.* [68] as the most fundamental texture feature. First, the coarseness is computed by taking the average $A_{k_t}(x,y)$ at every image pixel $I(x,y)$ over the neighborhood of size $2^{k_t} \times 2^{k_t}$ according to the following equation:

$$A_{k_t}(x,y) = \sum_{i=x-2^{k_t-1}}^{x+2^{k_t-1}-1} \sum_{j=y-2^{k_t-1}}^{y+2^{k_t-1}-1} \frac{f(i,j)}{2^{2k_t}} \quad (1)$$

where $f(x,y)$ represents the gray-level of image pixel $I(x,y)$ and $k_t \in [1,L]$ where $2^L \leq \min(W,H)$, W and H denote the effective width and height of the analyzed image. Second, at each pixel the differences $E_{k_t,h}(x,y)$ and $E_{k_t,v}(x,y)$ between the average of pairs corresponding to pairs of non-overlapping neighborhoods on opposite sides of the analyzed pixel in both the horizontal and vertical orientations, respectively, are computed as:

$$E_{k_t,h}(x,y) = |A_{k_t}(x+2^{k_t-1},y) - A_{k_t}(x-2^{k_t-1},y)| \quad (2)$$

$$E_{k_t,v}(x,y) = |A_{k_t}(x,y+2^{k_t-1}) - A_{k_t}(x,y-2^{k_t-1})| \quad (3)$$

Third, the best size $S_{best}(x,y) = 2^{k_t}$ is defined according to the specified k_t which maximizes $E = E_{max} = max_{1 \leq k_t \leq L}(E_{k_t,h}(x,y), E_{k_t,v}(x,y))$ in either the horizontal direction or the vertical one. Finally, the coarseness measure is defined as the average of S_{best} over the analyzed image according to the equation (9).
- Contrast (I_t^2, *cf.* equation (10)): This descriptor measures the dynamic range of gray-levels in an image with taking into consideration the distribution polarization of black and white pixels.
- Number of orientations (I_t^3, *cf.* equation (11)): This feature describes the local edge density and distribution of a texture. By building the histogram of local edge probabilities H_D against their directional angle, global texture features, such as long lines and simple curves, can be characterized. Firstly, the two following 3×3 operators (*cf.* equations (4) and (5)) are convolved with the input image to obtain the horizontal (Δ_H) and vertical (Δ_V) differences, respectively.

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (4) \qquad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (5)$$

Then, image edges can be detected by extracting the magnitude $|\Delta G|$ (*cf.* equation (6)) and direction θ_t (*cf.* equation (7)) at each pixel.

$$|\Delta G| = \frac{|\Delta_V| + |\Delta_H|}{2} \quad (6)$$

$$\theta_t = \tan^{-1}\frac{\Delta_V}{\Delta_H} + \frac{\pi}{2} \quad (7)$$

Therefore, H_D is produced by quantifying θ_t and counting all pixels respecting $|\Delta G| \geq t_H$ where t_H denotes the specified H_D threshold which is set to 12.
H_D is defined to be:

$$H_D(l) = \frac{N_{\theta_t}(l)}{\sum_{i=0}^{n_b-1} N_{\theta_t}(i)} \quad (8)$$

where n_b denotes the number of H_D bins which is set to 16. $l = 0, 1, \ldots, n_b - 1$. $N_{\theta_t}(l)$ is the number of pixels at which $\frac{(2l-1)\Pi}{2n_b} \leq \theta_t < \frac{(2l+1)}{2n_b}$.
Therefore, the number of orientations describes the local edge density and distribution which is given by extracting salient histogram peaks (*i.e.* local histogram maxima) after computing the difference vector between two successive histogram bins, according to the equation (11).

- Directionality (I_t^4, *cf.* equation (12)): This descriptor provides an insight into the global texture property over a region by measuring the total degree of texture directionality. It is computed by using an histogram of local edge probabilities H_D against their directional angle. By quantifying the sharpness of H_D peaks, the texture directionality is measured by summing the second moments around each peak according to the equation (12).

The Tamura features which have been investigated in this article, are summarized in Table 2.

Table 2: Tamura features.

Feature	Expression & Description
Coarseness	$I_t^1 = \frac{1}{WH} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} S_{best}(x,y)$ (9) This feature illustrates the scale and repetition rates of texture. Specifically, it measures the largest size at which a texture exists.
Contrast	$I_t^2 = \frac{\sigma}{\sqrt[4]{\alpha_4}}$ (10) This descriptor measures the dynamic range of gray-levels in an image with taking into consideration the distribution polarization of black and white pixels.
Number of orientations	$I_t^3 = \sum_k [\text{argmax}_{0 \leq k \leq n_b - 1} (\frac{\partial H_D(k)}{\partial k} = 0)]$ (11) This feature describes the local edge density and distribution of a texture.
Directionality	$I_t^4 = 1 - r\ n_p \sum_p^{np} \sum_{\Phi_h \in w_p} (\Phi_h - \Phi_p)^2 H_D(\Phi_h)$ (12) This descriptor provides an insight into the global texture property over a region by measuring the total degree of texture directionality.

In Table 2, σ and α_4 are the standard deviation estimator and the fourth root which was suggested by Tamura *et al.* [68] based on their experiments, respectively. α_4 is computed as:

$$\alpha_4 = \frac{\mu_4}{\sigma^4} \tag{13}$$

where μ_4 and σ represent the fourth moment about the mean μ. n_p, Φ_p, w_p, r and Φ_h denote the number of histogram peaks which was set by Tamura *et al.* [68] to 2, the p^{th} peak position of H_D, the range of p^{th} peak between valleys, the normalizing factor related to the quantized levels of Φ_h, and the quantized direction code (cyclically in modulo 2π), respectively.

3.2 LBP features

The LBP operator is one of the most explored local image descriptor for texture analysis which has mainly been used for describing local texture properties of images. It has been introduced to measure pure and original property of the texture spectrum by Wang and He [74]. They proposed a texture analysis pattern based on a texture unit. LBP is a two-level version of the texture spectrum method. Later, it was popularized by Harwood *et al.* [34] to analyze texture characteristics for texture classification.

LBP is obtained by locally thresholding texture and their combinations with local gray-scale measures. It represents each analyzed image pixel with a binary pattern based on the difference between its gray-level value and its circular neighborhood with specified radius R_l. If the gray-level value difference between the analyzed pixel $I_c(x,y)$ and its P_l neighboring pixels $I_{p\in[0,P_l-1]}(x,y)$, is greater than or equal to zero, the LBP value is set to 1, otherwise it is set to 0. Thus, the resistance to the intensity value of pixels in gray-scale format is ensured. If the coordinates of the analyzed pixel are $(0,0)$, then the coordinates of $I_p(x,y)$ are given by $(-R_l \sin(\frac{2\Pi p}{P_l}), R_l \cos(\frac{2\Pi p}{P_l}))$. The interpolation is applied when the gray-level values of neighbors mismatches to an image pixel integer value. Then, by multiplying the binary elements with a binomial coefficient, the LBP value $0 \leq LBP_{P_l,R_l}(I_c(x,y)) \leq 2^{P_l}$ which corresponds to the value of the LBP feature vector, is produced. The LBP operator LBP_{P_l,R_l} is defined according to the following equation:

$$LBP_{P_l,R_l}(I_c(x,y)) = \sum_{p=0}^{P_l-1} s(f_p(x,y) - f_c(x,y))2^p \tag{14}$$

where

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \tag{15}$$

where P_l is the number of neighboring pixels in a circular set. $f_{p\in[0,P_l-1]}(x,y)$ corresponds to the gray-level values of equally spaced pixels from $I_c(x,y)$ on a circle of radius R_l which builds the P_l circularly symmetric neighbors $I_{p\in[0,P_l-1]}(x,y)$. $f_c(x,y)$ and $f_p(x,y)$ represent the gray-levels of the analyzed image pixel $I_c(x,y)$ and image pixel $I_p(x,y)$, respectively.

By taking into account P_l pixels in the neighbor set when computing a basic LBP_{P_l,R_l} operator, 2^{P_l} different binary patterns are obtained. The obtained 2^{P_l} binary patterns are not rotationally invariant. Thereby, by performing a circular bitwise right-shift on the p-bit binary pattern and selecting the minimum value of $P_l - 1$ bit-wise right-shift operations on the binary pattern (*i.e.* assigning a unique identifier to each rotation invariant LBP), n_l unique rotation invariant local binary patterns are produced to remove the effect of rotation. Indeed, the quantification of the occurrence statistics of individual rotation invariant patterns corresponding to image micro-features is ensured. The rotation invariant LBP operator $LBP^{ri}_{P_l,R_l}$ is defined according to the following equation:

$$LBP^{ri}_{P_l,R_l}(I_c(x,y)) = \min_{0 \leq i \leq P_l-1} \{ROR(LBP_{P_l,R_l}(I_c(x,y),i))\} \tag{16}$$

where $ROR(.,i)$ represents a i times circular bit-wise right-shift on the P_l-bit binary pattern.

Noting that the obtained LBP feature vector is non-uniform, Ojala *et al.* [59] proposed an efficient multi-scale approach based on uniform local binary patterns for gray-scale and rotation invariant texture classification. They showed that the basic 3×3 LBP operator provides better performance by extracting uniform and non-uniform patterns from it. A pattern is considered as uniform, if the number of spatial transitions (bit-wise $0/1$ changes) in the pattern are less than or equal to 2. Therefore, the rotation invariant uniform 2 LBP operator is labeled "riu2". Formally, the rotation invariant uniform 2 LBP operator $LBP^{riu2}_{P_l,R_l}$ is defined according to the following equation:

$$LBP^{riu2}_{P_l,R_l}(I_c(x,y)) = \begin{cases} \sum_{p=0}^{P_l-1} s(g_p - g_c), \\ \text{if } U(LBP_{P_l,R_l}(I_c(x,y))) \leq 2 \\ P_l + 1, \text{otherwise.} \end{cases} \quad (17)$$

where

$$U(LBP_{P_l,R_l}(I_c(x,y))) = |s(g_{P_l-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P_l-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (18)$$

By using the $LBP^{riu2}_{P_l,R_l}$ operator, $P_l + 1$ uniform binary patterns are produced in a circularly symmetric neighbor set of P_l pixels. Each uniform binary pattern is labeled differently (*i.e.* a unique label is assigned to each uniform binary pattern corresponding to the number of "1" bits in the pattern $(0 \rightarrow P_l)$), while the non-uniform patterns are grouped in the "miscellaneous" label $P_l + 1$. Hence, by using the $LBP^{riu2}_{P_l,R_l}$ operator as gray-scale invariant measure of texture characteristics of an image, the distribution of the binary patterns for the whole analyzed image is described by computing the histogram of binary patterns H_{P_l,R_l}.

Recently, the LBP operator has gained great attention of many researchers in the DIA fields. LBP_{P_l,R_l}, $LBP^{ri}_{P_l,R_l}$ and $LBP^{riu2}_{P_l,R_l}$ features were extracted by Bhowmik and Kar [9] to localize text in HDIs. They used three LBP operators by setting R_l equal to 1, 2 and 3 and P_l equal to 8, 16 and 24, respectively. But, they considered only P_l equal to 8 during the binary pattern computation. They concluded that the obtained results of the three models (LBP_{P_l,R_l}, $LBP^{ri}_{P_l,R_l}$ and $LBP^{riu2}_{P_l,R_l}$) are relatively similar in most cases.

In this study, a rotation invariant uniform 2 LBP operator which is labeled $LBP^{riu2}_{P_l,R_l}$, is used. For describing an image with $LBP^{riu2}_{P_l,R_l}$, a histogram of binary patterns H_{P_l,R_l} of $P_l + 2$ bins is produced. Each bin provides an estimation of the probability to find the corresponding pattern in the analyzed image. P_l and R_l are set to 8 and 1, respectively. Thus, for each image pixel $I_c(x,y)$, $LBP^{riu2}_{P_l=8,R_l=1}(I_c(x,y))$ produces 10 $H_{P_l=8,R_l=1}$. The number of uniform and non-uniform patterns are 9 and 28, respectively, to ensure better discrimination of spatial patterns. Indeed, 10 $LBP^{riu2}_{P_l=8,R_l=1}$ descriptors are extracted. The $LBP^{riu2}_{P_l=8,R_l=1}$ feature vector consists of 10 terms of the probability to find the corresponding pattern in the analyzed image. The nine first descriptors correspond to the nine $H_{P_l=8,R_l=1}$ bins which represent the uniform patterns (*cf.* equation (19)), while the last one represents the last $H_{P_l=8,R_l=1}$ bin which characterizes all the non-uniform patterns (*cf.* equation (20)).

The $LBP^{riu2}_{P_l,R_l}$ features which have been investigated in this article, are summarized in Table 3.

Table 3: LBP features.

Feature	Expression & Description
Heights of the uniform bins of the histogram of binary patterns	$I^1_l = H_{P_l,R_l}(i)$ where $1 \leq i \leq P_l + 1$ (19) These features represent the uniform patterns.
Height of the non-uniform bin of the histogram of binary patterns	$I^2_l = H_{P_l,R_l}(i)$ where $i = P_l + 2$ (20) This descriptor characterizes all the non-uniform patterns.

In Table 3, H_{P_l,R_l} is the histogram of binary patterns. P_l is the number of neighboring pixels in a circular set of radius R_l.

3.3 GLRLM features

The GLRLM descriptors are extracted by applying the run-length method. The run-length method has been extensively studied in a wide array of fields for analysis of images and particularly for pattern recognition and texture classification [69]. It has been introduced by Galloway *et al.* [28] to classify a set of terrain samples by extracting various run-length features from several GLRLM.

For a given image, an element of the GLRLM $p(g,l)$ is defined as the number of runs with pixels of gray-level g and run-length l. A gray-level run g is a sequence in a scan direction of a set of consecutive and collinear image pixels with identical gray-level value. The length of the run l is the number of image pixels in the run. A GLRLM is computed for runs having any given direction. Usually, the four following scan directions have been used, $\theta_r = \{0, \pi/4, \pi/2, 3\pi/4\}$. For the GLRLM, the dimension of g is equal to G^l which corresponds to the maximum gray-level (*i.e.* number of gray-level bins), whereas the dimension of l is equal to L

which corresponds to the maximum run-length. Afterwards, a 2D run-length histogram ($H_{g,l}$) is produced for each scan direction, such one axis represented the run-length and the other axis illustrates the gray-level value or gray-level value bin. $H_{g,l}$ is a histogram of run-lengths. Therefore, since the $H_{g,l}$ is normalized, the probability of a specific run-length $P(g,l)$ can be defined according to the following equation:

$$\sum_{g=0}^{G^l-1}\sum_{l=1}^{L}P(g,l)=1 \tag{21}$$

where G^l is the number of gray-level bins (*i.e.* number of bins into which the image has been quantized) g is the gray-level value bin, L is the maximum run-length, and l is the run-length.

In this study, four 2D run-length histograms ($H_{g,l}$) are produced for each scan direction $\theta_r=\{0,\pi/4,\pi/2,3\pi/4\}$. For each 2D run-length histograms $H_{g,l}$, a feature vector of 11 terms of GLRLM indices is computed. The 11 texture features based on gray-level run-lengths and particularly the 2D run-length histogram ($H_{g,l}$) are introduced by Galloway *et al.* [28] to capture the coarseness of a texture in a specific direction.

The GLRLM features which have been investigated in this article, are summarized in Table 4.

Table 4: GLRLM features.

Feature	Expression & Description
Short-run emphasis	$I_r^1=\sum_{g=0}^{G-1}\sum_{l=1}^{L}\frac{P(g,l)}{l^2}$ (22) This metric ensures the characterization of fine-grained textures by emphasizing short runs.
Long-run emphasis	$I_r^2=\sum_{g=0}^{G-1}\sum_{l=1}^{L}P(g,l)l^2$ (23) This feature helps to characterize textures with large homogeneous areas or coarse textures by emphasizing long runs.
Low gray-level emphasis	$I_r^3=\sum_{g=0}^{G-1}\sum_{l=1}^{L}\frac{P(g,l)}{(g+1)^2}$ (24) This measure is orthogonal to SRE (*cf.* equation 22) and it provides an insight of the dominance of many runs of low gray-level value in the analyzed texture.
High gray-level emphasis	$I_r^4=\sum_{g=0}^{G-1}\sum_{l=1}^{L}P(g,l)(g+1)^2$ (25) This measure is orthogonal to LRE (*cf.* equation 23) and it provides information on the dominance of many runs of high gray-level value in the analyzed texture.
Gray-level non-uniformity	$I_r^5=\sum_{l=1}^{L}[\sum_{g=0}^{G-1}P(g,l)]^2$ (26) This metric is focused on detecting the gray-level outliers from the histogram.
Run-length non-uniformity	$I_r^6=\sum_{g=0}^{G-1}[\sum_{l=1}^{L}P(g,l)]^2$ (27) This metric is an indicator of few run-length outliers which are dominating the histogram.
Run percentage	$I_r^7=\sum_{g=0}^{G-1}\sum_{l=1}^{L}\frac{1}{P(g,l)l}$ (28) This metric gives a glimpse into the overall histogram homogeneity. The maximum RPC value corresponds to the case where all runs are equal to the unity length regardless of the gray-level values.
Short-run low gray-level emphasis	$I_r^8=\sum_{g=0}^{G-1}\sum_{l=1}^{L}\frac{P(g,l)}{l^2(g+1)^2}$ (29) This measure is a combination of the two metrics: SRE (*cf.* equation 22) and LGRE (*cf.* equation 24) which estimates the dominance of many short runs of low gray-level value.
Long-run high gray-level emphasis	$I_r^9=\sum_{g=0}^{G-1}\sum_{l=1}^{L}P(g,l)l^2(g+1)^2$ (30) This feature is the complementary metric to SRLGE (*cf.* equation 29). It characterizes the combination of long high gray-level value runs.
Short-run high gray-level emphasis	$I_r^{10}=\sum_{g=0}^{G-1}\sum_{l=1}^{L}\frac{P(g,l)(g+1)^2}{l^2}$ (31) This measure is both orthogonal to SRLGE (*cf.* equation 29) and LRHGE (*cf.* equation 30). It carries out the domination of short runs with high intensity gray-levels in the analyzed texture.
Long-run low gray-level emphasis	$I_r^{11}=\sum_{g=0}^{G-1}\sum_{l=1}^{L}\frac{P(g,l)l^2}{(g+1)^2}$ (32) This feature is the complementary metric to SRHGE (*cf.* equation 31). It allows to characterize long runs with low intensity gray-levels in the analyzed texture.

Continued on next column . . .

Continued from previous column

In Table 4, $P(g,l)$ corresponds to the probability of a specific run-length, g is the gray-level value $H_{g,l}$ bin, and l is the run-length.

3.4 Auto-correlation features

The auto-correlation features are extracted from a non-parametric tool which consists of the auto-correlation function. The auto-correlation function which is a 2D function, is defined as a similarity measure between a dataset and a shifted copy of the data. It is used to find periodic and similar patterns through a number of extracted auto-correlation features [62]. The auto-correlation function which is computed along the horizontal and vertical axes of the analysis window of an image I, is defined according to the following equation:

$$\begin{aligned} R^{I(\alpha,\beta)}_{(x,y)} &= \sum_{\alpha\in\Omega}\sum_{\beta\in\Omega} I(x,y)I(x+\alpha,y+\beta) \\ &= FFT^{\text{-}1}\left[FFT\left[I(x,y)\right]FFT^{*}\left[I(x,y)\right]\right] \end{aligned} \quad (33)$$

where $I(x+\alpha,y+\beta)$ is the translation of the analysis window of an image $I(x,y)$ by α and β pixels along the horizontal and vertical axes, respectively, defined on the plane Ω. FFT, $(.)^{*}$ and $(.)^{\text{-}1}$ denote the fast Fourier transform, complex conjugate and inverse transform, respectively.

By analyzing the auto-correlation results, a polar diagram which is called a rose of directions can be produced. The rose of directions reveals the significant orientations of the texture in the analyzed image block. It highlights interesting information concerning the principal orientations and periodicities of the texture, characterizing the content of images without any assumption about page structure and its characteristics. The rose of directions has recently been used with HDIs [40]. In order to identify the main orientation of the analyzed image, the rose of directions is computed for each orientation by summing up the different values of the auto-correlation function (*cf.* equation (33)):

$$R^{I}_{(x,y)}(\Theta_i) = \sum_{D_i} R^{I(\alpha,\beta)}_{(x,y)} \quad (34)$$

where $\Theta_i \in [0,180]$ is the selected orientation of the set of possible orientations D_i which is represented by a straight line passing through (x,y) and the angle Θ_i.

The rose of directions is normalized in order to select only the relative variations of all contributions for each direction [40]. The relative sum $R'^{I}_{(x,y)}(\Theta_i)$ is defined as:

$$R'^{I}_{(x,y)}(\Theta_i) = \frac{R^{I}_{(x,y)}(\Theta_i) - R^{I}_{min}}{R^{I}_{max} - R^{I}_{min}} \quad (35)$$

where $R^{I}_{max} \neq R^{I}_{min}$, R^{I}_{min} and R^{I}_{max} represent the minimum and maximum values of $R^{I}_{(x,y)}(\Theta_i)$, respectively. Both are computed on the analysis window of an image $I(x,y)$.

The auto-correlation features which have been investigated in this article, are summarized in Table 5 [40,51].

Table 5: Auto-correlation features.

Feature	Expression & Description
Main angle of the rose of directions	$I^1_a{=}\lvert 180{-}\text{argmax}_{\Theta_i\in[0,180]}(R'^{I}_{(x,y)}(\Theta_i))\rvert$ (36) This metric ensures the characterization of the main orientation of a texture.
Intensity of the auto-correlation function for the main orientation	$I^2_a{=}R^{I}_{(x,y)}(\text{argmax}_{\Theta_i\in[0,180]}(R'^{I}_{(x,y)}(\Theta_i)))$ (37) This feature helps to characterize the anisotropy of a texture.
Variance of the intensities of the rose of directions	$I^3_a{=}\sigma^2_a(R'^{I}_{(x,y)}(\Theta_i))$ (38) This measure provides an insight of the overall shape of the rose of directions.
Mean stroke width along specific directions	$I^4_a{=}\Sigma_{\Theta\in[10,80]}\lvert I(x,y){-}T^{\Theta}_{(\alpha,0)}(I(\frac{y}{\lvert\tan(\Theta)\rvert},y))\rvert$ (39) This measure estimates the mean stroke width along specific directions.
Mean stroke height along specific directions	$I^5_a{=}\Sigma_{\Theta\in[10,80]}\lvert I(x,y){-}T^{\Theta}_{(0,\beta)}(I(x,x\lvert\tan(\Theta)\rvert))\rvert$ (40) This metric corresponds to the estimation of mean stroke height along specific directions.

Continued on next column . . .

Continued from previous column

In Table 5, the standard deviation estimator σ_a is computed as:

$$\sigma^2_a = \frac{1}{\theta_a - 1}\sum_{i=1}^{\theta_a}(R'^{I}_{(x,y)}(\Theta_i))^2 - \frac{\theta_a}{\theta_a - 1}\mu^2_a \quad (41)$$

where μ_a denotes the mean value of the intensities of the rose of directions. θ_a is the possible number of the orientation values of the rose of directions (*i.e.* 179 orientation values). $R'^{I}_{(x,y)}(\Theta_i)$ is a normalization of the rose of directions.

3.5 GLCM features

The GLCM or co-occurrence matrix is a classic of statistical texture-based segmentation methods. The GLCM is an estimate of the second order probability density function of image pixels. This matrix determines the probability of occurrence of pixel pairs according to their gray-levels and distance by considering the spatial relationship of pixels in the image [33].

A GLCM element is the probability of the gray-level pairs defined in a specified direction θ_c and separated by a particular distance of d_c units. The co-occurrence descriptors are then statistics computed from the GLCM. They provide second order statistical information of neighboring pixels of an image. Multi-distance and multi-direction can be applied to extract a large number of GLCM descriptors. Usually, the co-occurrence matrices are generated for a small range of distance values $d_c = \{1,2\}$ and typically for the directions $\theta_c{=}\{0,\pi/4,\pi/2,3\pi/4\}$ [11].

In this study, from the computed co-occurrence matrices, eight GLCM features are extracted for two distances $d_c = \{1,2\}$ [11]. In addition to the 16 co-occurrence features (8 for each distance), two other descriptors are computed, mean value (*cf.* equation (50)) and standard deviation (*cf.* equation (51)) of the energy, for the two combined distances [48]. Busch *et al.* [11] showed that the 18 selected and extracted GLCM features perform well for script identification.

The co-occurrence features extracted from the GLCM which have been investigated in this article, are summarized in Table 6.

Table 6: GLCM features.

Feature	Expression & Description
Maximum probability	$I_c^1=\max_{i,j}\left\{p_{(d_c,\theta_c)}(i,j)\right\}$ (42) This metric ensures the record of the highest GLCM element. High values of GLCM element will occurred if one combination of pixels dominates pixel pairs.
Correlation metric	$I_c^2=\sum_{i=0}^{255}\sum_{j=0}^{255}\frac{(i-\mu_r)(j-\mu_c)p_{(d_c,\theta_c)}(i,j)}{\sigma_r\sigma_c}$ (43) This feature helps to measure the gray-level linear dependence between pixels at the specified positions relative to each other. It has a large value when the values are uniformly distributed in the GLCM and a low value otherwise.
Energy	$I_c^3=\sum_{k=0}^{255}D(k)$ (44) This measure which has also been called angular second moment, provides an insight of image homogeneity. It has low value when the probabilities of the gray-level pairs have very similar values and a high value otherwise.
Entropy	$I_c^4=-\sum_{k=0}^{255}D(k)\log_2 D(k)$ (45) This metric characterizes the energy values for pixel combinations. It measures the disorder or randomness of the GLCM. Inhomogeneous texture have low first order entropy, while a homogeneous texture has a high entropy.
Contrast	$I_c^5=\sum_{k=0}^{255}k^2D(k)$ (46) This metric which has also been called inertia, corresponds to a measure of the contrast by computing a difference moment of the GLCM and it estimates the contrast or it quantifies local variation present in the analyzed image.
Local homogeneity	$I_c^6=\sum_{k=0}^{255}\frac{D(k)}{1+k^2}$ (47) This measure has also been called inverse difference moment. It is higher when we find the same pair of pixels which is in the case that the gray-level is uniform or when there is a spatial periodicity.
Cluster shade	$I_c^7=\sum_{i=0}^{255}\sum_{j=0}^{255}(i-\mu_r+j-\mu_c)^3p_{(d_c,\theta_c)}(i,j)$ (48) This metric corresponds to a measure of the gray-level distribution around the mean, with a high ability to discriminate the third order. It measures the skewness of the GLCM (*i.e.* lack of symmetry). When it is high, the analyzed image is not symmetric.
Cluster prominence	$I_c^8=\sum_{i=0}^{255}\sum_{j=0}^{255}(i-\mu_r+j-\mu_c)^4p_{(d_c,\theta_c)}(i,j)$ (49) This metric corresponds to a measure of the gray-level distribution around the mean, with a high ability to discriminate the fourth order. It also measures the skewness of the GLCM.
Energy mean	$I_c^{17}=\sum_{k=0}^{510}kD(k)$ (50) This metric corresponds to the mean of the energy feature computed from the two distance values $d_c=1,2$.
Energy standard deviation	$I_c^{18}=\sqrt{\sum_{k=0}^{510}(k-I_c^3)^2D(k)}$ (51) This metric corresponds to the standard deviation of the energy feature computed from the two distance values $d_c=1,2$. It characterizes the uniformity of the texture when varying the specified distance.

Continued on next column ...

Continued from previous column

In Table 6, $p_{d_c,\theta_c}(i,j)$ is the probability of the gray-level pair i and j defined in a specified direction θ_c and separated by a particular distance of d_c units.

$$p_r(i)=\sum_{i=0}^{255}p_{d_c,\theta_c}(i,j) \qquad p_c(j)=\sum_{j=0}^{255}p_{d_c,\theta_c}(i,j)$$

$$\mu_r=\sum_{i=0}^{255}p_r(i) \qquad \mu_c=\sum_{j=0}^{255}p_c(j)$$

$$\sigma_r^2=\sum_{i=0}^{255}i^2p_r(i)-\mu_r^2 \qquad \sigma_c^2=\sum_{j=0}^{255}j^2p_c(j)-\mu_c^2$$

$$D(k)=\sum_{\substack{0\leq i\leq 255\ \ 0\leq j\leq 255\\ |i-j|=k}}p_{d_c,\theta_c}(i,j)$$

3.6 Gabor features

The Gabor features are extracted using the multi-channel Gabor filtering technique. The original Gabor elementary functions have been firstly proposed by Gabor [27]. The multi-channel Gabor filtering is inspired by the multi-channel filtering theory which has been first investigated by Campbell and Robson [12] for the visual information processing of the human visual system. Daugman [22] modeled the visual information processing of the human visual system by the 2D multi-channel Gabor functions which are local spatial band-pass filters. The main idea of the multi-channel filtering technique is to exploit the differences in dominant sizes and orientations of different textures by decomposing the original image into several filtered images with limited spectral information. The 2D Gabor functions have the advantage to have the conjoint resolution information in both the 2D spatial and Fourier domains. The filtered images are proceeded by tuning the analyzed image to combinations of frequency and orientation in a narrow range which are referred to channels and interpreted as band-pass filters. By applying a bank of Gabor filters, the specified channels cover the spatial frequency domain.

A 2D Gabor filter is a linear selective band-pass filter, dependent on two parameters (spatial frequency f_g and orientation θ_g) which characterize the specified channel. It consists of a Gaussian kernel function modulated by a sinusoidal plane wave. The spatial frequency f determines the distance from the Gaussian centers to the origin while the orientation θ_g specifies the angle from the horizontal axis (*i.e.* α-axis to the Gaussian centers). The multi-channel Gabor filtering approach is inherently multi-resolutional which is a close relative of the wavelet transform [38].

The Gabor transform of an image $I(x,y)$ is:

$$I_{G_{(f_g,\theta_g)}}(x,y) = \sum_{\alpha\in\Omega}\sum_{\beta\in\Omega} I(x+\alpha, y+\beta)\; G_{(f_g,\theta_g)}(\alpha,\beta) \qquad (52)$$

where f_g and θ_g are the spatial frequency and orientation of the Gabor filter envelope.

$$G_{(f_g,\theta_g)}(\alpha,\beta) = \sqrt{[G_{e(f_g,\theta_g)}(\alpha,\beta)]^2 + [G_{o(f_g,\theta_g)}(\alpha,\beta)]^2}$$

$$G_{e(f_g,\theta_g)} = \frac{H_{1(f_g,\theta_g)}(\alpha,\beta) + H_{2(f_g,\theta_g)}(\alpha,\beta)}{2}$$

$$G_{o(f_g,\theta_g)} = \frac{H_{1(f_g,\theta_g)}(\alpha,\beta) + H_{2(f_g,\theta_g)}(\alpha,\beta)}{2j}$$

$$H_{1(f_g,\theta_g)}(\alpha,\beta) = \exp\{-2\pi\sigma_g^2[(\alpha - f_g\cos\theta_g)^2 + (\beta - f_g\sin\theta_g)^2]\}$$

$$H_{2(f_g,\theta_g)}(\alpha,\beta) = \exp\{-2\pi\sigma_g^2[(\alpha + f_g\cos\theta_g)^2 + (\beta - f_g\sin\theta_g)^2]\}$$

$$j^2 = -1$$

where $G_{e(f_g,\theta_g)}$ and $G_{o(f_g,\theta_g)}$ denote the spatial frequency responses of the even- and odd-symmetric Gabor filter. σ_g denotes the space constant of the Gabor filter envelope.

The four directions (0, $\pi/4$, $\pi/2$ and $3\pi/4$) and the six spatial frequencies ($2\sqrt{2}$, $4\sqrt{2}$, $8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$ and $64\sqrt{2}$) are widely used in the literature [39, 77]. In this study, the magnitude response of the output of Gabor functions is investigated. The magnitude of the output is important if the specified Gabor filter matched the particular texture, otherwise low response to the specified Gabor filter corresponds to poor match of the dominant texture properties of the analyzed image to the set of the spatial frequency components of the fixed Gabor filter [10]. 24 Gabor filters are applied (6 different spatial frequencies f_g=$\{2\sqrt{2}, 4\sqrt{2}, 8\sqrt{2}, 16\sqrt{2}, 32\sqrt{2},\ 64\sqrt{2}\}$ and 4 different orientations θ_g=$\{0, \pi/4, \pi/2, 3\pi/4\}$). The space of Gabor filter is set to $\sigma_g = \sigma_x = \sigma_y = 1$. When convolving an image with 24 Gabor channels (obtained by using 6 different spatial frequencies and 4 different orientations), 24 Gabor-filtered images are produced. In this study, 24 Gabor responses are generated. Finally, by convolving the analyzed whole DI at each specified channel defined by a pair of spatial frequency and orientation, the Gabor features are extracted from the magnitudes of the Gabor-filtered images. The extracted Gabor features represent the statistical distribution of the Gabor magnitude response.

The two simple first order statistics representing the Gabor features which have been investigated in this article, are detailed in Table 7.

Table 7: Gabor features.

Feature	Expression & Description
Mean of the Gabor-filtered magnitude responses	$I_g^1 = \frac{\Sigma_{x=1}^{M_g}\Sigma_{y=1}^{N_g} I_{G_{(f_g,\theta_g)}}(x,y)}{M_g N_g}$ (53) This feature characterizes the average of the Gabor filtered magnitude response corresponding to all pixels defined in the analyzed sliding window of the filtered image. This descriptor quantifies how the dominant texture properties of the analyzed image match to the set of spatial-frequency components of the fixed Gabor filter.
Standard deviation of the Gabor-filtered magnitude response	$I_g^2 = \frac{\Sigma_{x=1}^{M_g}\Sigma_{y=1}^{N_g} [I_{G_{(f_g,\theta_g}}(x,y) - F_{(f_g,\theta_g)}^{(1)}]^2}{M_g N_g}$ (54) This descriptor determines how much the dispersion from the computed mean of the Gabor filtered magnitude response exists.

In Table 7, M_g and N_g denote the width and height of the Gabor-filtered magnitude response, respectively.

3.7 Wavelet features

Mallat [49] investigated the application of the wavelets as multi-resolution representations to data compression in image coding, texture discrimination and fractal analysis. The wavelet features which are extracted from the wavelet transform provide interesting insight into the statistical characteristics of the analyzed image. The wavelet features represent consistent properties in the localization of the spatial frequency and multi-resolution. A 2D wavelet transform ensures the localization in both the scale domain (*i.e.* frequency) via dilations and in the time domain via translations of the mother wavelet. A 2D wavelet transform represents an image with both the spatial and frequency characteristics. The 2D wavelet decomposition is processed by using a high-pass filter g^f, a low-pass filter h^f and a 2D scaling function ϕ. It assumes the three following wavelet functions (ψ):

$$\begin{cases} \psi^{(I)}(x,y) = \phi(x)\psi(y) = 2\sum_{k,l} g^{(I)}(k,l)\phi(2x-k,2y-l) \\ \psi^{(II)}(x,y) = \psi(x)\phi(y) = 2\sum_{k,l} g^{(II)}(k,l)\phi(2x-k,2y-l) \\ \psi^{(III)}(x,y) = \psi(x)\psi(y) = 2\sum_{k,l} g^{(III)}(k,l)\phi(2x-k,2y-l) \end{cases} \tag{55}$$

where

$$\begin{cases} g^{(I)}(k,l) = & h^f(k)g^f(l) \\ g^{(II)}(k,l) = & g^f(k)h^f(l) \\ g^{(III)}(k,l) = & g^f(k)g^f(l) \end{cases} \tag{56}$$

The objective of a 2D wavelet transform is to decompose an image into low and high frequency sub-band images (*i.e.* to filter out several frequency ranges). The 2D J-level wavelet transform decomposes a discrete input image $I(x,y)$ into 4 sub-bands and it produces $3J+1$ sub-images:

$$A_{2^{-J}}, \{D^{(v)}_{2^{-j}}, D^{(h)}_{2^{-j}}, D^{(d)}_{2^{-j}}\}_{j=1,2,\ldots,J} \tag{57}$$

where J represents the scale of the discrete wavelet transform. j denotes the decomposition level of the discrete wavelet transform such as $j = 1,2,\ldots,J$. $A_{2^{-J}}$ is the approximation of the input image $I(x,y)$ at 2^{-J} resolution. $D^{(v)}_{2^{-j}}$, $D^{(h)}_{2^{-j}}$ and $D^{(d)}_{2^{-j}}$ are 3 detail components of the input image $I(x,y)$ at 2^{-j} resolution. The wavelet coefficients in $D^{(v)}_{2^{-j}}$, $D^{(h)}_{2^{-j}}$ and $D^{(d)}_{2^{-j}}$ illustrate the vertical, horizontal and diagonal high frequencies, respectively.

The approximation (*cf.* equation (58)) and detail (*cf.* equation (59)) coefficients are computed according to the following equations:

$$C^{Aj}_{k,l} = \int_{-\infty}^{+\infty} 2^j \phi(2^j x - k, 2^j y - l) f_s(x,y) \text{ dx dy} \tag{58}$$

$$C^{D(s)j}_{k,l} = \int_{-\infty}^{+\infty} 2^j \psi^{(s)}(2^j x - k, 2^j y - l) f_s(x,y) \text{ dx dy} \tag{59}$$

where $D(s)j$ denotes the vertical, horizontal or diagonal detail components of the input image $I(x,y)$ at 2^{-j} resolution. $f_s(x,y)$ represents the pixel gray-level of a sub-band or sub-image from the 2D wavelet decomposition.

The Haar and Daubechies wavelets are the most used ones since they have been shown to work effectively in numerous applications. The Haar wavelet transform is the fastest among all wavelets since its coefficients are either 1 or -1. Thus, they are the less complex and most widely used wavelets. The Daubechies ones are characterized by the fractal structures. The distribution characteristics of the wavelet coefficients of the 1-level Haar transform was investigated for DI segmentation. The results confirmed that the performance produced by the two longer wavelet filters (4-tap Daubechies and 8-tap Daubechies) was similar while the Haar transform had the best localization property since its filter was the shortest and it had the lowest processing time [47]. Therefore, in this study the wavelet features are extracted from three different 2D 3-level discrete stationary wavelet transform with a limited number of taps: 3-level wavelet transform using Haar filter (Haar), 3-level wavelet transform using 3-tap Daubechies filter (Db3) and 3-level wavelet transform using 4-tap Daubechies filter (Db4). Therefore, 10 sub-bands ($A_{2^{-3}}$, $D^{(v)}_{2^{-1}}$, $D^{(h)}_{2^{-1}}$, $D^{(d)}_{2^{-1}}$, $D^{(v)}_{2^{-2}}$, $D^{(h)}_{2^{-2}}$, $D^{(d)}_{2^{-2}}$, $D^{(v)}_{2^{-3}}$, $D^{(h)}_{2^{-3}}$ and $D^{(d)}_{2^{-3}}$) are generated.

In our experiments, in order to reduce the number of wavelet coefficients, two simple statistics deduced from the wavelet transform coefficients for each sub-band are extracted to form a feature vector of 20 terms (10 sub-bands).

The wavelet features which have been investigated in this article, are summarized in Table 8.

Table 8: Wavelet features.

Feature	Expression & Description
Mean of the wavelet transform coefficients	$I^1_w = \frac{\Sigma_{i=0}^{S_w} \Sigma_{j=1}^{S_h} C(i,j)}{S_w S_h}$ (60) This feature characterizes the average of the wavelet transform coefficients for each sub-band defined in the analyzed sliding window of the image. This descriptor represents the average of 2-D signal in various frequency bands.
Standard deviation of the wavelet transform coefficients	$I^2_w = \frac{\Sigma_{i=0}^{S_w} \Sigma_{j=1}^{S_h} [C(i,j) - F^{(1)}]^2}{S_w S_h}$ (61) This descriptor determines how much the dispersion from the computed mean of wavelet transform coefficients exists.

In Table 8, $C(i,j)$ is the transform wavelet coefficient. S_w and S_h are the width and height of a sub-band in the wavelet domain, respectively.

3.7.1 Haar

The Haar wavelet employs a low-pass filter h^f_{Haar} and a high-pass filter g^f_{Haar}.

where $h^f_{Haar} = [\sqrt{2}, \sqrt{2}]$ and $g^f_{Haar} = [-\sqrt{2}, \sqrt{2}]$.

3.7.2 Db3

The Db3 wavelet employs a low-pass filter h^f_{Db3} and a high-pass filter g^f_{Db3}.

where

$$h^f_{Db3} = [0.0352, -0.0854, -0.1350, 0.4598, 0.8068, 0.3326]$$

$$g^f_{Db3} = [-0.3326, 0.8068, -0.4598, -0.1350, 0.0854, 0.0352]$$

3.7.3 *Db*4

The Db4 wavelet employs a low-pass filter h^f_{Db4} and a high-pass filter g^f_{Db4}.

where

$$h^f_{Db4} = [-0.0105, 0.0328, 0.0308, -0.1870, \\ -0.0279, 0.6308, 0.7148, 0.2303]$$

$$g^f_{Db4} = [-0.2303, 0.7148, -0.6308, -0.0279, \\ 0.1870, 0.0308, -0.0328, -0.0105]$$

4 Experimental protocol

In order to assess the discriminating power and determine the computational cost of the nine investigated texture-based feature sets, previously presented in Section 3, a wide variety of HDIs and different HDI content types have been selected and a classical pixel-labeling scheme is proposed. In this section, a brief description of the main phases of the pixel-labeling scheme used for comparing texture features is presented. Subsequently, the performance of each texture-based feature set is detailed after describing our experimental corpus and its associated ground truth, and presenting the used accuracy metrics for performance evaluation.

4.1 Pixel-labeling scheme for comparing texture features

The extraction of texture descriptors helps to describe the DI layout and content by analyzing the texture feature space computed from the extracted textural characteristics of DI content (*i.e.* by mapping the differences in the spatial structures of each digitized DI into differences in gray-level values for each page). However, different results are shown according to the specified extracted kind of texture used for segmenting or characterizing the DI layout on the one hand, and the DI content on the other hand. Therefore, in this study we aim at determining the performance of each texture feature set according to the DI content and providing an additional insight into the computational cost of each analyzed texture feature set. However, there is a real need for a generic and standard framework that permits a fair comparison of texture features. For this purpose, a classical pixel-labeling scheme for comparing texture features is proposed in this study (*cf.* Figure 1). This scheme is considered as the support of this comparative study or benchmarking of the nine different texture-based feature sets. Since we aim at characterizing a wide variety of HDI contents and layouts by analyzing the textural properties of HDI contents, a pixel-based approach is adopted in this study due to its advantage to overcome the limits and constraints of region and boundary-based approaches [19].

The pixel-labeling scheme used in our experiments to compare different texture features, is illustrated in Figure 1. It is conceptualized by three modular processes:

1. ***Pre-processing and foreground pixel selection*** (*cf.* Section 4.1.1),
2. ***Texture feature extraction*** (*cf.* Section 4.1.2),
3. ***Pixel-clustering and labeling*** (*cf.* Section 4.1.3).

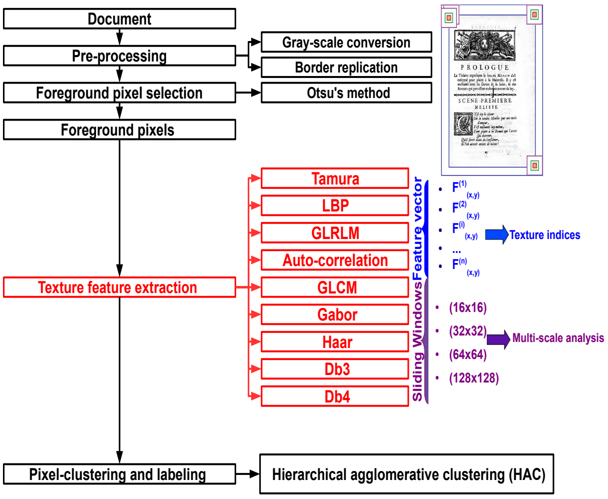


Fig. 1: Pixel-labeling scheme for comparing texture features.

The pixel-labeling scheme can be considered as a prerequisite step in the pipeline of DIA and particularly region segmentation and classification. However, due to a possible bias produced by performing a classification task, this step is not included in this study by applying a training phase through supervised machine learning tools. Therefore, the pixel classification and post-processing tasks are beyond the scope of this comparative study. The pixel-labeling task is necessary for further data processing by different techniques since it provides the basis for all subsequent segmentation, analysis, classification and recognition processes. Indeed, the pixel-labeling phase is considered as the first major step in a pixel-based DIA workflow after the image preprocessing.

4.1.1 *Pre-processing and foreground pixel selection*

First, a HDI is fed as input and is read as a gray-scale image. The extraction of texture information is processed on gray-scale images without introducing any binarization step.

Then, to deal with pixels at image borders when computing texture features on the whole image, a border replication step is introduced. Furthermore, in order to reduce data cardinality and obtain a significant gain in the computation time and used memory, the texture descriptors are extracted only on the foreground pixels. It is worth noting that the foreground texture is more interesting to categorize HDIs, as it represents the main information of their contents and layouts. Moreover, the foreground pixel selection step does not give rise any specific problem related to the loss of information as long as the background texture information is considered when texture features are extracted from the foreground pixels by means of a pixel-based approach based on a multi-scale technique (*cf.* Section 4.1.2). Therefore, the texture descriptors are extracted only on the foreground pixels.

Several comparative studies of text/background segmentation or binarization methods for degraded HDIs had been reviewed [35]. These studies do not agree on the best method and none has been shown to be perfect and suitable for HDIs, even not local binarization approaches. Nevertheless, Gatos *et al.* [30] demonstrated superior performance of a proposed adaptive approach for the binarization and enhancement of degraded documents compared with four well-known binarization techniques even when the documents are very noisy and highly degraded. Other studies suggested modifying the existing binarization techniques in order to retain their advantages and modify them to retrieve the grayscale information. For instance, Villegas *et al.* [72] proposed a modified binarization algorithm based on the well known Sauvola thresholding technique for handwritten text recognition [64]. They compared the performances of their algorithm with those of two state-of-the-art techniques: the classical Sauvola thresholding and the background estimation and subtraction method [63]. They concluded firstly that their algorithm outperformed the previously used methods based on background estimation and subtraction. In addition, an improvement in recognition performance was noted since the grayscale information was preserved. However, the proposed method required a specific parameter tuning by the user. Moreover, it was evaluated on a single corpus which is the “ESPOSALLES” database [63].

Our study focuses on assessing the classification of the selected foreground pixels. Even when few pixels are may be missing, a texture-based approach is still robust (*i.e.* analyzing all pixels is not very useful). In this work, the foreground pixel selection step is performed using a standard parameter-free binarization method, the Otsu’s method, to retrieve only those pixels representing information of the foreground (*i.e.* noise, text, graphics, *etc.*) [61]. However, using the Otsu’s method is beyond the scope of this work, good results have been observed when using it. For instance, Busch *et al.* [11] used Otsu’s method to segment and extract the text regions from a DI. Using a global thresholding approach, Otsu’s method has provided an adequate and fast mean of binarization to retrieve only the foreground pixels and subsequently extract texture features from only the selected foreground pixels. As an example, for a full historical page document (1965×2750 pixels), scanned at 300 dpi, the number of the foreground pixels is equal to 26086. Thus, the rate of the foreground pixels is lower than $\frac{1}{200}$.

4.1.2 Texture feature extraction and multi-scale analysis

The texture feature extraction is performed using a pixel-wise technique, *i.e.* by using analysis windows of varying sizes in order to adopt a multi-resolution/multi-scale approach. The pixel-wise technique is chosen since it gives more reliable values and ensures more accurate determination of texture boundary, however it has a high demand in memory and computational time. Furthermore, using a multi-scale approach in DIA fields and pyramid methods in several image processing applications [17], rich information (e.g. gray-level distribution) can be produced since textural characteristics can be perceived differently at varying scales.

The sizes of sliding windows vary classically from (16×16) to (256×256) in the existing pixel-based methods using a multi-scale analysis in order to overcome the large variability of the sizes and resolutions of the analyzed HDIs [40]. However, the computation time is highly dependent on the resolution, size of the analyzed DI and number of the foreground pixels. As a matter of fact, in this study the sizes of sliding windows vary from (16×16) to (128×128), because beyond the (128×128) size the step of the texture feature extraction would be both too costly and time-consuming. In addition, using a large size of a sliding window misleads an observation with coarse texture expression. Hence, the optimal size of each sliding window is determined respecting a constructive compromise between the computation time and the pixel-labeling quality (reliable measurement and texture boundary). The sliding window is shifted horizontally and vertically to scan the whole HDI. Therefore, a feature vector is computed on a foreground pixel-per-pixel basis. Each pixel is represented by scalar texture features, determined according to a small region bounded by contour of the analyzed sliding window. The analyzed sliding window is centered on that pixel. Subsequently, the extracted texture indices from the foreground pixels are aggregated into the N^f-dimensional (N^f-D) array on pixel-by-pixel basis, where N^f represents the number of extracted texture indices by applying multi-scale analysis.

4.1.3 Pixel-clustering and labeling

The goal of this step is to structure the texture feature space within a clustering technique in order to group pixels sharing similar characteristics and to identify and characterize unlabeled data (obtained from the texture feature extraction phase). The partition and analysis task of the set of unlabeled data into groups or clusters is necessary to extract from the analyzed DI regions which have homogeneous characteristics and similar properties with respect to the extracted texture features. This task is considered as a feature space structuring technique.

Since an unsupervised pixel-labeling scheme for comparing texture features is applied in this study, an unsupervised clustering step is needed to group pixels sharing similar characteristics. For instance, Nguyen *et al.* focused their study on specific graphics called drop caps and particularly on the extraction of shapes in these graphics, as part of an attempt to provide wider access to historical collections [56]. They found interesting classification results which were obtained by performing the hierarchical agglomerative clustering (HAC) algorithm on the stroke features of drop caps [46]. Therefore, the HAC algorithm is chosen in the proposed pixel-labeling scheme for comparing texture features.

HAC processes by successively merging pairs of existing clusters where at each cluster grouping step, the choice of cluster pairs depends on the smallest distance (*i.e.* clusters are grouped if the intra-cluster inertia is minimal). Therefore, in this study each pixel is automatically assigned to one of a number of possible clusters according to the contents of its feature vector by applying the HAC algorithm on the normalized texture features and setting the maximum number of homogeneous and similar content regions equal to the one defined in the ground truth. The texture feature vectors are normalized to zero mean and unit standard deviation in order to avoid a domination of the higher numerical range of a few features. By partitioning texture-based feature vector sets into compact and well-separated clusters in the feature space, individual pixels are labeled without taking into account the spatial coordinates which lead to the application of the pixel-clustering and labeling steps, producing a pixel-labeled image as output. As a matter of fact, the spatial information is also not integrated in the pixel-labeling scheme for comparing texture features, to avoid bias caused by introducing a refinement pixel-labeling phase with taking into consideration the topological relationships of pixels. In addition, the number of homogeneous and similar content regions has been set to the one defined in the ground truth when performing the HAC algorithm in the pixel-labeling scheme for comparing texture features. The aim is to avoid inconsistencies and bias in assessments caused by estimating automatically the number of homogeneous and similar content regions and subsequently to ensure an objective understanding of the behavior of the evaluated texture feature sets.

4.2 Corpus and preparation of ground truth

Many important issues arise to provide an informative benchmarking of the most classical and widely used texture-based feature sets for HDI layout analysis and HDI segmentation such as the lack of a common dataset of HDIs and the lack of the appropriate quantitative evaluation measures for the segmentation quality [67]. Moreover, many researchers have addressed the need of a good dataset. Antonacopoulos *et al.* [1] considered a dataset as a good one if it is realistic (*i.e.* it must be composed of real digitized DIs), comprehensive (*i.e.* it must be well characterized and detailed for ensuring in-depth evaluation) and flexibly structured (*i.e.* to facilitate a selection of sub-sets with specific conditions). Although the issue of the realistic dataset availability and the broadband access to researchers for the performance evaluation of contemporary DIs have been discussed and solved by Antonacopoulos *et al.* [1], representative datasets of HDIs with their associated ground truths are currently hardly publicly accessible for HDI layout analysis. Finding a large corpus of HDIs having many annotated HDIs with various content and layout characteristics and which were collected from several European libraries is still a challenging issue for HDI layout analysis. This is mainly due to the intellectual and industrial property rights. Another challenge facing founding a representative dataset of HDIs concerns the definition of its objective and complete associated ground truth. Defining an objective ground truth is still not a straightforward task due to their characteristics (e.g. noise and degradation, presence of handwriting, overlapping layouts, great variability of page layout). These characteristics complicate the definition of the appropriate and objective ground truth, the characterization or segmentation of HDIs and make the processing of this kind of DIs a difficult task.

The different datasets of HDIs provided in the context of the contests of the ICDAR and the ICFHR conferences and the HIP workshop focus on either a specific kind of document such as historical newspaper layout analysis (HNLA), or a specific application such as handwritten text recognition (HTRtS and RHHT), multi-spectral text extraction (MS-TEx), text line detection (ANDAR-TL), word recognition (ANWRESH), keyword spotting (KWS), classification of medieval handwritings in Latin script. The main datasets provided in the context of these competitions are composed of pages having similar content and layout characteristics or collected from a single book or collection such as the GERMANA corpus or the RODRIGO corpus [65]. Indeed, there is a limited number of realistic, comprehensive and flexibly structured datasets of HDIs and their associated ground-truths for HDI layout analysis. Recent page segmentation, historical document layout analysis and historical book recognition (HBR) contests in the context of the ICDAR conference and the HIP workshop have provided a dataset (2011 and 2013) [2, 3]. This dataset, which is called the *HBR2013 dataset* in this article, is a subset of the IMPACT dataset, representing key holdings of major European libraries and consisting of printed documents of various types (e.g. books, newspapers, journals, legal documents), in 25 languages from the 17^{th} century to the early 20^{th} century. It represents a wide variety of layouts that reflect several particularities of HDIs. It focuses on the complete recognition workflow for books, comprising different scenarios such as layout analysis (page segmentation and region classification) and text recognition (OCR). It is composed of 100 binary, gray-scale or color HDIs which were digitized at 150/300 dpi. It was selected as it has as little as possible artifacts (e.g. severe page curl, arbitrary warping) to overcome the use of a separate image enhancement step before the DI layout analysis task. The ground truths of only six pages have been provided.

Our experimental corpus is composed of the two datasets, the *DIGIDOC-Texture dataset* (*cf.* Figure 2) and the *HBR2013 dataset* (*cf.* Figure 3).

In our experiments, we firstly collected 1000 real scanned HDIs from 298 books in different languages and scripts from the 13^{th} century to the early 20^{th} century. In this study, this dataset is called the *DIGIDOC-Texture dataset*. The *DIGIDOC-Texture dataset* contains 1000 ground truthed one-page HDIs which have been collected from Gallica[1]. The HDIs of the *DIGIDOC-Texture dataset* were selected from several books across a variety of disciplines, such as novels, law texts, educational books (e.g. history, geography, nature) and xylographic booklets, to provide a broader range of HDI contents and layouts. The selected HDIs are gray-scale/color DIs which were digitized at 300/400 dpi and saved in the TIFF format which provides a high resolution of digitized images. The 1000 HDIs of the *DIGIDOC-Texture dataset* has been structured into four categories of real scanned HDIs differentiated by their content (*cf.* Figure 2 and Table 9), reflecting the challenges of this study to determine which texture features can be more adequate for segmenting the graphical contents from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand. The *DIGIDOC-Texture dataset* contains a sufficient number of images with both simple and complex layouts for each category of HDIs which have been ground truthed to ensure a better understanding of the behavior of the evaluated texture feature sets.

To study the scalability of the nine evaluated texture-based feature sets in another HDI corpus, a standard “public” and representative dataset of HDIs and its associated ground truth are considerably necessary for our experiments. Therefore, we are constrained by carrying out our experiments on the 100 HDIs if the *HBR2013 dataset*. The *HBR2013 dataset* was provided in the context of the ICDAR conference and the HIP workshop (2011 and 2013) by the IMPACT research team[2]. It can be considered as a complex one and different from the *DIGIDOC-Texture dataset* for the following reasons. First, there is a large diversity of the HDI contents of the *HBR2013 dataset* compared to the *DIGIDOC-Texture dataset* (*i.e.* the values of the number of types of content regions defined in the ground truth of the *HBR2013 dataset* vary from 2 to 6). Second, it contains binary images. Finally, some images were digitized at low resolution, which might potentially introduce a bias in the texture feature extraction and analysis tasks. In our experiments, we have structured the 100 HDIs of the *HBR2013 dataset* into nine different categories differentiated by their content (*cf.* Figure 3 and Table 10). The six different categories of the *HBR2013 dataset* do not contain sufficient number of images. The number of images in each category of the *HBR2013 dataset* vary from 3 to 20 (*cf.* Table 10) .

Table 9: Composition of the *DIGIDOC-Texture dataset*.

Content	**Number of pages**	**Number of fonts**	**Graphics**
Graphics and one text font (*cf.* Figure 2(a))	250	1	Yes
Graphics and text with two different fonts (*cf.* Figure 2(c))	250	2	Yes
Only two fonts (*cf.* Figure 2(e))	250	2	No
Only three fonts (*cf.* Figure 2(g))	250	3	No

Table 10: Composition of the *HBR2013 dataset*.

Content	**Number of pages**	**Number of fonts**	**Graphics**
Only one font (*cf.* Figure 3(a))	3	1	No
Only two fonts (*cf.* Figure 3(c))	17	2	No
Graphics and text with two different fonts (*cf.* Figure 3(e))	9	2	Yes
Only three fonts (*cf.* Figure 3(g))	20	3	No
Graphics and text with three different fonts (*cf.* Figure 3(i))	6	3	Yes
Only four fonts (*cf.* Figure 3(k))	11	4	No
Graphics and text with four different fonts (*cf.* Figure 3(m))	15	4	Yes
Only five fonts (*cf.* Figure 3(o))	5	5	No
Graphics and text with five different fonts (*cf.* Figure 3(q))	14	5	Yes

The characteristics of our experimental corpus are primarily: strong heterogeneity, with differences in layout, typography, illustration style, historic fonts, complex layouts (e.g. dense printing, irregular spacing, varying text column widths, marginal notes), ink shining through and historical spelling variants, *etc.* In addition to this specificity, the issues affecting DI layout analysis, such as the degradation properties (e.g. yellow pages, ink stains, back-to-front interference) and scanning defects (e.g. defects of curvature and light) are adequately covered. It is worth noting that the analyzed images for this study were selected so as to be as realistic as possible, in order to reflect the challenges of this work to determine if the evaluated texture features are sufficiently robust to the particularities of HDIs. Figure 4 il-

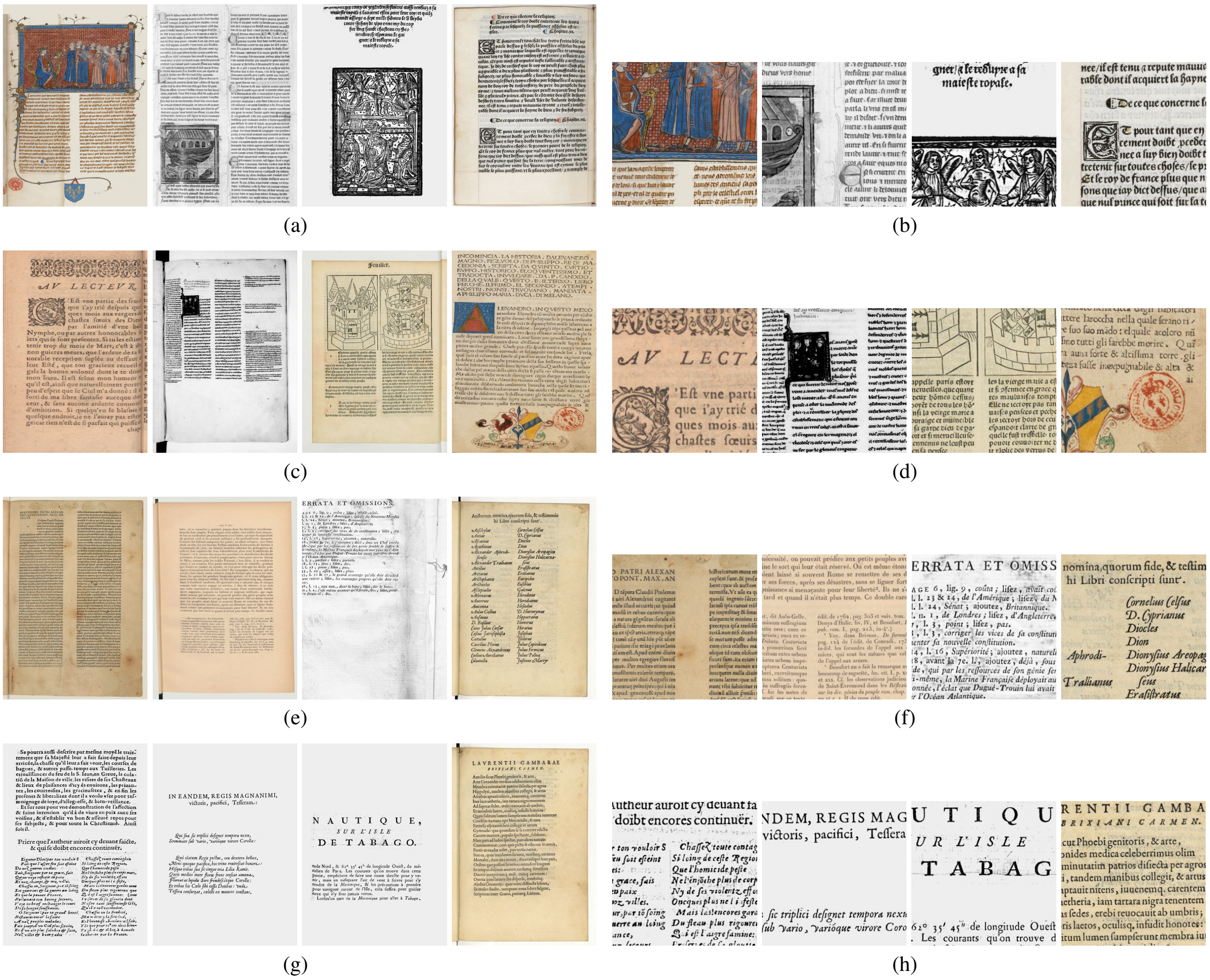

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 2: HDI examples of the *DIGIDOC-Texture dataset*. Figures (a), (c), (e) and (g) illustrate examples of HDIs of the *DIGIDOC-Texture dataset* containing graphics and one font, graphics and two fonts, only two fonts and only three fonts, respectively. Figures (b), (d), (f) and (h) depict respectively zoomed regions of the examples of HDIs of the four categories of the *DIGIDOC-Texture dataset*.

lustrates some particularities of the evaluated HDIs in our experiments. These particularities are related to the digitization process such as page skew, scanning defects (curvature, light, blur), presence of black borders, *etc*.

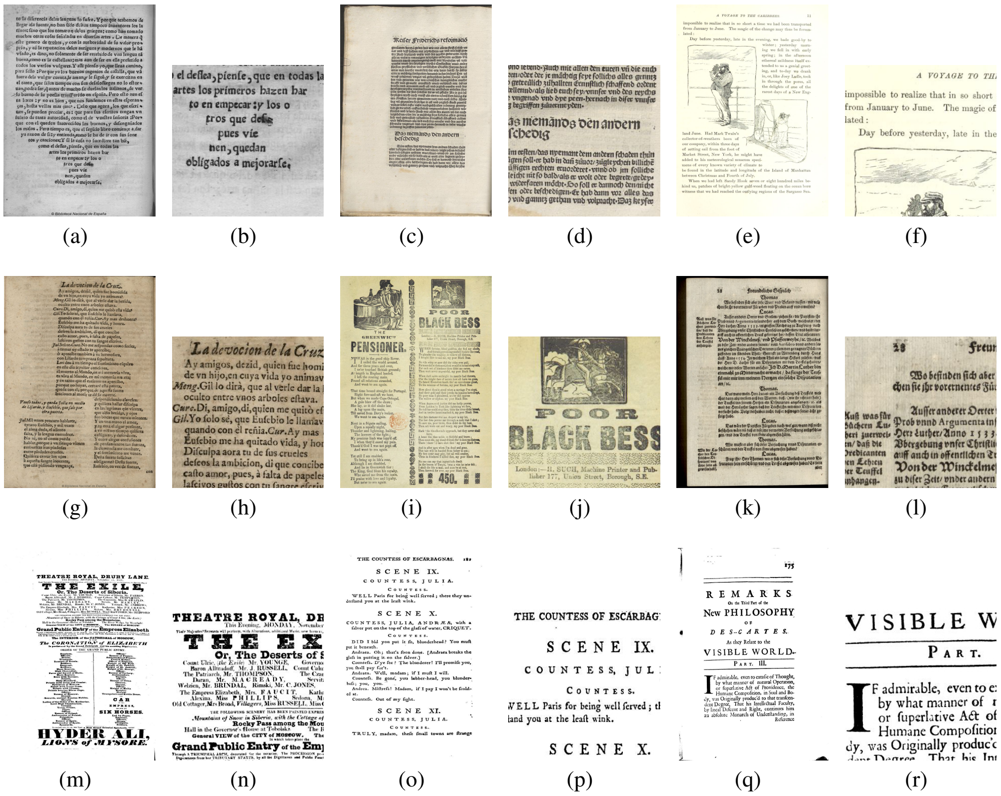
For the two datasets, *DIGIDOC-Texture dataset*[3] and *HBR2013 dataset* (*cf.* Figures 2 and 3), the ground truth has been manually outlined using rectangular regions drawn around each selected zone. The regions have been ground truthed by zoning each content type (*i.e.* each rectangular region has been classified into text or graphics). Different labels for regions with different fonts have been also assigned for evaluating the performance of texture feature to separate various text fonts (*cf.* Figure 5). The ground truth has been produced using the ground truthing editor, ground truthing environment for document images (GEDI)[4], a public domain DI annotation tool that labels spatial boundaries of regions.

5 Experiments and results

To analyze and evaluate the robustness of the nine investigated texture feature sets and provide additional insights into their classification accuracy, and computational cost (*i.e.* memory requirements, processing time, numerical complexity, and texture vector dimensionality), an informative benchmark of the performance and computational

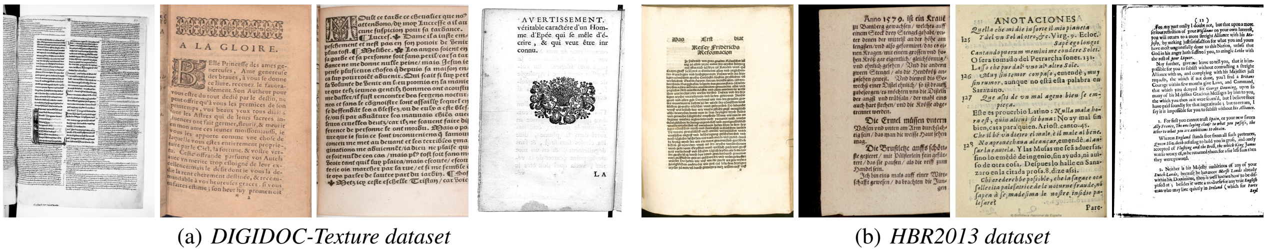
[3] The *DIGIDOC-Texture dataset* and its ground truth are temporarily available on `http://litis-digidoc.univ-rouen.fr/texture/DIGIDOC-Texture.tar.gz`. This dataset is available on request subject to the agreement from the French national library "bibliothèque nationale de France" (BnF).

[4] `http://gedigroundtruth.sourceforge.net/`



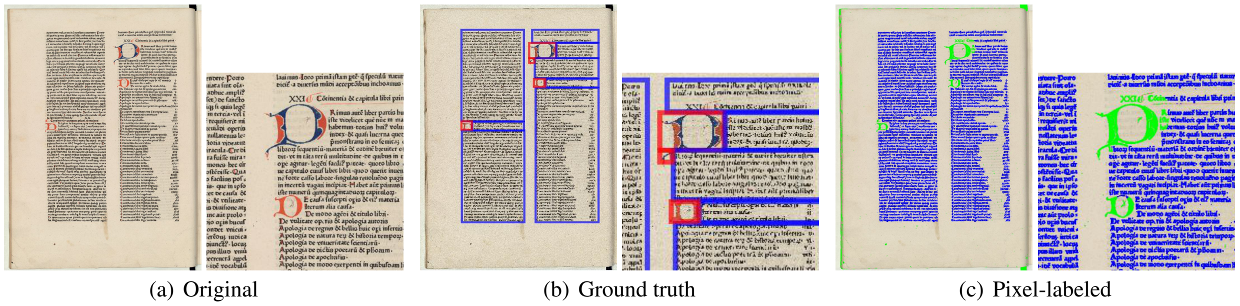
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o) (p) (q) (r)

Fig. 3: HDI examples of the *HBR2013 dataset*. Figures (a), (c), (e), (g), (i), (k), (m), (o) and (q) illustrate examples of HDIs of the *HBR2013 dataset* containing only two fonts, two fonts and graphics, only three fonts, three fonts and graphics, only four fonts, four fonts and graphics, only five fonts and five fonts and graphics, respectively. Figures (b), (d), (f), (h), (j), (l), (n), (p) and (r) depict respectively zoomed regions of the examples of HDIs of the nine categories of the *HBR2013 dataset*.



(a) *DIGIDOC-Texture dataset* (b) *HBR2013 dataset*

Fig. 4: Illustration of some particularities of the evaluated HDIs in our experiments (e.g. page skew, scanning defects, presence of black borders). Figures (a) and (b) show examples of HDIs which were selected from the *DIGIDOC-Texture dataset* and the *HBR2013 dataset*, respectively.



(a) Original (b) Ground truth (c) Pixel-labeled

Fig. 5: Example of the defined ground truth and the obtained pixel-labeling result. Figure (a) illustrates an original HDI and a zoomed region. Figure (b) depicts its associated ground truth. Figure (c) shows the final result of the pixel-labeling task by analyzing the Gabor features.

cost of each texture-based feature set is firstly given. Qualitative and numerical experiments are presented to analyze each texture-based feature set performance (*cf.* Section 5.1). Moreover, a correlation analysis of the performance of each texture-based feature set is proposed to show the similarities of the behavior of the different evaluated texture features (*cf.* Section 5.2). In addition, a statistical comparison of the performances of the nine analyzed texture-based feature sets in this study has been proposed to validate the obtained results over two different datasets (*cf.* Section 5.3). Besides, based on the experimental results, many observations and recommendations about the choice of texture features which are well suited for segmenting different content types are detailed. We finally conclude by identifying texture features which represent a constructive compromise between the computational cost and the pixel-level labeling quality (*cf.* Section 5.4).

5.1 Benchmarking

In this section, a comparative study or a benchmarking of the nine previously presented texture-based feature sets in Section 3, is detailed based on the obtained performance using the proposed pixel-labeling scheme for comparing texture features (*cf.* Figure 1). First, the computational cost is presented by providing an additional insight into the processing time and complexity of each texture-based feature set (*cf.* Section 5.1.1). Qualitative and quantitative evaluations on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset*, have been conducted to analyze each texture-based feature set performance for assessing pixel-level labeling quality in Sections 5.1.2 and 5.1.3.

5.1.1 Computational cost

The benchmarking of the nine investigated texture-based approaches in this study has been run on a SGI Altix ICE 8200 cluster (one central processing unit and 2 GB allocated memory on a Quad-Core $X5355@2.66GHz$ running on Linux), without a determined and focused effort to achieve an optimized implementation of the investigated texture-based features. Analyzing the nine sets of texture descriptors using the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset*, gives a total of 9900 analyzed images ($1000+100$ images $\times$ 9 different texture-based approaches).

The scalar features are extracted separately from the nine texture-based feature sets using four different sliding window sizes (*cf.* Section 3 and Table 13). In this study, we are interested in raising issues related only to how these texture-based sets are compared with each other. We avoid bias caused by introducing a feature selection task, such as the methods based on a dimension reduction technique. Moreover, it is quite certain that a feature selection task can not be adapted to all kinds of HDIs since the texture indices can have different ranges from a HDI to another one and from a corpus to another one.

An additional insight into the computational cost (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality) is provided in Table 13. The processing time highly depends on the resolution, size of the input image and number of the foreground pixels. An example of the computational cost of extracting and analyzing the nine investigated texture-based feature sets in this study from a full page document scanned at 300 dpi (1965×2750 pixels) is illustrated in Table 13. The highest time required to process this page is obtained when using the wavelet approaches while the lowest one is obtained when using the GLCM descriptors (*i.e.* it is reduced to only 14 seconds). The computation time of each texture feature set is in concordance with its complexity. We can see that the Db4-based approach has the highest complexity while the lowest one is noted for the GLCM-based approach (*cf.* Table 13). Therefore, this study states the GLCM-based approach is the best one in terms of processing time and complexity. However, the GLCM-based and the Gabor-based approaches are the highest memory-consuming (*i.e.* more than 587 MB used memory). We note that even if the three investigated wavelets consume a similar amount of memory, they have different computation times. The Haar-based approach is the best one among the three investigated wavelets in terms of computational cost. This confirms that the Haar wavelet transform is the fastest among the examined wavelets (*cf.* Section 3.7). However, the auto-correlation and LBP-based approaches have similar computational cost, they have different feature dimensions (*i.e.* the dimension of the LBP feature vector is the double of the auto-correlation one). Nevertheless, we observe the increase of the feature dimension of the Gabor and GLRLM-based approaches (*i.e.* the Gabor and the GLRLM signatures correspond to a set of vectors composed of 192 and 176 numerical values, respectively).

5.1.2 Qualitative results

The results of applying each texture feature set to many examples of HDIs are shown in Figures 7, 8, 9, 10, 11 and 12. Since the process is unsupervised, the colors attributed to text or graphics may differ from one HDI to another. A visual comparison of the resulting images using the proposed pixel-labeling scheme for comparing texture features on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset* is illustrated in Figure 7. By visual inspection of the obtained pixel-labeled HDIs, we note that most of the investigated texture-based approaches provide satisfying results particularly in distinguishing the textual regions from the graphical ones (*cf.* Figure 7).

– *DIGIDOC-Texture dataset*

In Figures 7(a) to 7(e) where the analyzed HDI contains one text font and graphics, the pixel-labeling results given by analyzing the texture-based feature sets on the proposed pixel-labeling scheme are relatively similar and satisfying in distinguishing the textual regions from the graphical ones. The pixel-labeling results of the investigated texture feature sets show a significant discriminating power for separating text (single font) and graphic regions when comparing visually the segmentation results. Nevertheless, by comparing the visual results given by the nine investigated texture-based feature sets, we note that the graphic regions (green) are more homogeneous when using Gabor features (*cf.* Figure 7(d)) than when using the other texture features. However, the Gabor features have more difficulty separating textual regions (blue) when they are too spatially close to the graphical ones (*i.e.* textual regions which are spatially close to the graphic ones have been mis-labeled).

We observe that the GLCM-based, the Gabor-based and the Db4-based approaches perform considerably better in segmenting documents containing only textual regions with distinct fonts by distinguishing two different text fonts, the handwritten notes in the margins and the printed text (*cf.* Figures 7(h), 7(i) and 7(j)).

In Figure 8, where the HDI under consideration contains two fonts and graphics, the nine investigated sets of texture features can not separate properly textual regions with different sizes and fonts. By analyzing the most sets of texture features for the *"Two fonts and graphics"* category of HDIs, two clusters are produced for graphic regions by discriminating the noise on the HDI borders. This points out that the texture features have also more difficulty segmenting two distinct text fonts when the involved HDI contains graphics.

We show that the Gabor features are the best in segregating three different fonts, text with S_1^f size font (red), text with $S_2^f \neq S_1^f$ size font (blue) and italic (green) fonts in Figure 9(f). This may be confirmed by the frequent use of the Gabor descriptors mainly to identify script and language and for character and font recognition in the literature [14, 77], since the Gabor features are known to be sensitive to the stroke width. Indeed, Gabor filters have the advantage to present the optimal localization properties for capturing information in both the spatial and frequency domains from the analyzed HDIs (*i.e.* Gabor filters are inherently multi-resolutional). On the other side, for the other texture features including the three investigated kinds of wavelets (*cf.* Figures 9(a) to 9(e) and 9(g) to 9(i)), the outcomes are poorer in segregating three different fonts.

We note that the wavelet-based approaches and more specifically Db3 and Db4, perform slightly similarly to the Gabor one and particularly in the case of HDIs containing graphics and text (*cf.* Figure 10). However, in certain cases the Gabor-based approach confuses the uppercase text and the graphical components unlike the wavelet-based approach. This confusion can be explained by the limitations of the Gabor approach to separate spatially close distinct kinds of information (*i.e.* the vertical/horizontal spacing is too small). Indeed, the Gabor features are extracted for a specified range of frequency and direction values. Thus, the performance of the Gabor approach depends directly on the layout document. Nevertheless, when using the Gabor primitives, we can see that distinct kinds of graphics can be discriminated (*cf.* Figures 10(g)).

We also observe that the GLRLM features are more appropriate to distinguish the textual regions from the graphical ones (*cf.* Figures 7(a) and 8(c)), but they have more difficulty separating textual regions (*cf.* Figures 7(f) and 9(c)). This is due to the fact that the GLRLM features do not accurately characterize a complex texture since they are based on analyzing simple indices (computed by means of the number of runs with pixels of a specific gray-level, predefined run-length and four standard orientations). The GLRLM features characterize mainly the coarseness level of a texture based on computing gray-level run-lengths by the means of the 2D run-length histogram (*cf.* Table 4).

– *HBR2013 dataset*

In an example of the *"Two fonts and graphics"* category of the *"HBR2013 dataset"* HDIs (*cf.* Figures 7(k) to 7(o)), we see that two clusters for graphic regions are obtained by discriminating many orientations that are present to different extents in graphic blocks. This confirms that the analyzed texture descriptors generally provide the main orientation of a texture. Therefore, the analyzed features have more difficulty segmenting two distinct text fonts when the documents also contain graphics. This strengthens our previous observations deduced when analyzing the *"Two fonts and graphics"* category of HDIs in the *DIGIDOC-Texture dataset* that the analyzed features in this study have also more difficulty segmenting two distinct text fonts when the documents also contain graphics. We conclude that most investigated texture feature sets can not separate properly textual regions with different sizes and fonts and particularly when the documents also contain graphics. A suitable alternative is to use recursive clustering methods in order to ensure the distinction between distinct text fonts and various graphic types when the documents under consideration are complex and contains graphics and various kinds of fonts. We also note that the Gabor features give the best results in terms of the homogeneity of the textual region content (*cf.* Figure 7(n)).

The results of applying the GLRLM, the auto-correlation, the GLCM, the Gabor and the Db4 features to an example of the *"Three fonts and graphics"* category of the *"HBR2013 dataset"* HDIs are illustrated in Figures 7(p) to 7(t). We also show that the Gabor features give the best results in terms of the homogeneity of the textual region content (*cf.* Figure 7(s)). A cluster representing the uppercase text font (blue) is clearly identified when analyzing the Gabor features on Figure 7(s). However, a slight confusion is also observed between the pixels of the uppercase text font (blue) and the graphical regions (green) (*cf.* Figure 7(s)).

Unlike the HDIs containing two fonts and graphics (*cf.* Figures 7(n) and 7(o)), the pixel-labeling results of an example of by analyzing the Db4 wavelet features (*cf.* Figure 7(t)) are not similar to those obtained by analyzing the Gabor descriptors when the HDI contains three fonts and graphics (*cf.* Figure 7(s)). This can be justified by the particularities of the analyzed HDI which is complex and contains graphics and various kinds of fonts. Besides, numerous spacing variations due to different font sizes and styles within one HDI can be seen. Hence, the wavelet-based approach fails to have homogenous regions because it does not cover a large range of frequency and direction values like the Gabor-based approach.

In the case of a HDI containing only textual regions with two different fonts (*cf.* Figure 11), we observe that the Gabor features are the best in segregating two different fonts, *i.e.* we distinguish two different text fonts, text with S_1^f size font (green) and text with $S_2^f \leq S_1^f$ size font (blue) (*cf.* Figure 11(f)). On the other side, the other investigated texture features have not borne the desired goal of segregating two different fonts. This strengthens our previous results obtained for the *DIGIDOC-Texture dataset* and confirms our assumption that the Gabor descriptors are the most suitable for font segmentation, since they are known to be sensitive to the stroke width.

In Figure 12, we observe that all investigated texture features even the Gabor features have failed to separate text fonts when the analyzed HDI contains only three different text fonts. This may be explained by the fact that the analyzed HDI has a copyright notice at the bottom of the page. This copyright notice has introduced artificial texture information and subsequently a bias in the texture feature extraction and analysis tasks.

5.1.3 Performance evaluation

However, comparing visually the effectiveness of a texture-based method is inherently a subjective evaluation and is not sufficient (*cf.* Section 5.1.2). Thus, it is necessary to assess quantitatively the results in order to have a conclusion of which set of texture features is well suited for firstly segmenting graphical regions from textual ones, and then for discriminating text in a variety of situations of different fonts and scales. Cote and Albu [19] confirmed that conducting qualitative and quantitative evaluations is crucial for analyzing texture-based methods used in DIA. Nevertheless, finding appropriate quantitative accuracy metrics is required first to evaluate the performance of the obtained results of the proposed pixel-labeling scheme for comparing the nine investigated texture feature sets.

In this study, we are more interested in characterizing a wide variety of HDI contents and layouts by analyzing the textural properties of HDI contents and finding homogeneous or similar content regions defined by similar texture indices. Thus, based on the pixel-accurate representation of page segmentation using the defined ground truth (*cf.* Section 4.2), several performance metrics have been computed to assess the behavior and the quality of a texture-based pixel-labeling algorithm. The performance of the different texture features have been analyzed by computing several per-pixel and per-block accuracy metrics, such as silhouette width (SW), purity per-block (PPB) and F-measure (F).

The silhouette width (SW) which is considered as an internal or unsupervised per-pixel accuracy metric, measures the level of compactness and separation by analyzing the distribution of the observations into clusters. For assessing the quality of segmenting historical document images into homogeneous or similar regions defined by similar texture indices, Mehri *et al.* [51] defined an external or supervised evaluation metric, called purity per-block (PPB). The PPB computes the homogeneity rate of regions by assessing the matching regions between the defined ground truth and the obtained pixel-labeling results. Finally, the F-measure (F) is computed in this study to get an insight into the per-pixel classification accuracy. It assesses both the homogeneity and the completeness criteria of the pixel-clustering and labeling results.

The performance evaluation and comparison of the nine investigated texture-based feature sets in this study using the proposed pixel-labeling scheme on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset*, are presented in Table 11. SW, PPB and F are computed. The higher the values of the computed metrics, the better the results. Measures of SW, PPB and F are presented at the bottom of each image in this article.

By analyzing the F-measure standard deviations of the nine investigated texture-based feature sets in this study using the proposed pixel-labeling scheme on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset* (*cf.* Table 12), variable values have been noted. The values of the F-measure standard deviations vary from 0.02 to 0.21. This shows that the texture-based feature sets perform very well on some HDIs, while failing on some other HDIs.

– ***DIGIDOC-Texture dataset***

In Table 11, the computed clustering and classification accuracy values are congruent. However, we note a slight difference in the performance of the SW average and a small variability in the ranking of the different investigated texture-based feature sets when computing the SW metric. This is due to the progressive merge process of the HAC algorithm used in the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets, where in higher levels in the hierarchy, two distant data points can be merged together and yet still belong to the same cluster after cutting the dendrogram. This causes a slightly lower value of the *SW*. This justification can be strengthened by the particularity of the *SW* as internal or unsupervised accuracy clustering evaluation which investigates the coherence of a clustering solution by measuring how observations are close to the cluster center and how clusters are well-separated.

We also observe that the best results of mean F values are obtained by the Gabor features for almost HDI categories of the *DIGIDOC-Texture dataset* (88%, 67%, 84% and 64% for the *"One font and graphics"*, the *"Two fonts and graphics"*, the *"Only two fonts"* and the *"Only three fonts"* HDI categories, respectively). Similarly, the best results of mean *PPB* values are observed when analyzing the Gabor features. This strengthens our previous observations obtained when analyzing visually the results (*cf.* Figures 7(d), 7(i), 8(f) and 9(f)).

In addition, we note that the second best performance is obtained for almost all HDI categories of the *DIGIDOC-Texture dataset* when using one of three investigated kinds of wavelet features on the proposed pixel-labeling scheme. This is due to the consistent properties of the wavelet features in the localization of the frequency space and multiresolution. We observe that the wavelet-based approaches and more specifically the Db4 wavelet one (*cf.* Figure 7(e)), perform quite similar to the Gabor one.

Low values of performance difference of the computed evaluation metrics between the used Gabor and wavelet features on the proposed pixel-labeling scheme when HDIs containing graphics and text (F difference values of 4% and 4% for the *"One font and graphics"* and the *"Two fonts and graphics"* HDI categories, respectively) compared to the case when HDIs containing only text (F difference values of 8% and 5% for the *"Only two fonts"* and the *"Only three fonts"* HDI categories, respectively). We conclude that the Gabor-based approach performs considerably better than the wavelet one if the analyzed HDI contains only text. Nevertheless, the values of the computed accuracy metrics are low with the *"Only three fonts"* category ($0.31SW$, $88\%PPB$ and $64\%F$ are noted when using the Gabor-based approach) comparing with the *"One font and graphics"*. As a consequence, the Gabor-based approach performs significantly better than the other investigated features specifically when the involved HDI contains two different text fonts or graphics and text. This strengthens our previous observations obtained when analyzing visually the results and confirms our assumption that the Gabor descriptors are the most suitable for font segmentation, since they are known to be sensitive to the stroke width.

We also observe that the performance values of the computed accuracy metrics for almost all HDI categories of the *DIGIDOC-Texture dataset* when using the auto-correlation descriptors are close to those when using the Gabor and wavelet features. This can be justified by the interesting information about the main orientation of a texture provided by the auto-correlation features, and which would ensure a relevant discrimination of the different classes of the foreground layers.

Overall, the worst performances are obtained for most of the computed evaluation metrics ($82\%PPB$ and $53\%F$ for the *"Overall"* category) when using the GLRLM features on the proposed pixel-labeling scheme. This strengthens our previous observations obtained when analyzing visually the results (*cf.* Figures 7(f) and 9(c)). For HDIs containing only distinct fonts, we observe that the lowest values of the computed clustering and classification accuracy metrics are divided among multiple texture-based feature sets (e.g. Tamura, GLRLM and GLCM descriptors). Therefore, we conclude that despite the lower computational cost of the Tamura, the GLRLM and the GLCM features, they are not adequate for separating different text fonts. This is due to the fact that these features are not sensitive to the stroke width to discriminate different font sizes and various font styles.

– ***HBR2013 dataset***

An important dimension that should be emphasized is that the participating methods in the ICDAR2013 competition on historical book recognition (HBR2013) using the *HBR2013 dataset* are analyzed according to two pre-defined scenarios: identify and label regions, and text recognition [3]. Moreover, the participating methods are mainly based on analyzing connected components. To adjust several parameters and thresholds, the participating methods are based on strong *a priori* knowledge such as the repetitiveness of document structure (e.g. blocks shape, uniformity in horizontal and/or vertical spacings and/or assumptions about textual and graphical characteristics such as font size). Therefore, the placement of the participating methods into context by comparing them to our results is not possible since this study is deliberately confined to the pixel-labeling task, which is considered as the first major step in a pixel-based DIA workflow and also the basis for all subsequent segmentation, analysis, classification and recognition processes.

In Figure 11, we observe that calculating the overall accuracy metrics on the *HBR2013 dataset* confirms the results obtained by using the *DIGIDOC-Texture dataset*. The Gabor-based approach is the best one (overall values of $91\%PPB$ and $51\%F$ are noted). This strengthens our previous observations obtained when analyzing visually the results (*cf.* Figures 7(n), 7(s) and 11(f)). However, we note a significant drop in performance ($25\%F$) when applying the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *HBR2013* *dataset* comparing the *DIGIDOC-Texture dataset*. This can be explained by the complexity of the *HBR2013 dataset*. Indeed, the values of the number of types of content regions defined in the ground truth are distributed in the interval of $[2,6]$ range. Unlike the *HBR2013 dataset*, the values of the number of types of content regions defined in the ground truth of the *DIGIDOC-Texture dataset* is equal to either 2 or 3. This highlights that it might be better to first discriminate text from graphic regions and then separate the different text fonts by means of recursive clustering methods to have better performance when the analyzed HDIs are complex and contains graphics and various kinds of fonts. Moreover, this confirms our observation about the slight difference in the performance of the *SW* and points out that the main technological bottleneck is the definition of an accurate and objective ground truth by determining fairly the number of different HDI content types. Thus, the performance of the results depends on the values of the number of types of content regions defined in the ground truth. The smaller values of the number of types of content regions defined in the ground truth represent higher efficiency. We note that the performance decreases since the number of text fonts increases. We also observe a significant difference in the *SW* performance comparing the two other computed accuracy metrics, *PPB* and F. This strengthens our previous observation about the particularity of the *SW* metric (*i.e.* the *SW* metric is sensitive to the number of different HDI content types).

We note that the second best performance is obtained for almost all HDI categories of the *HBR2013 dataset* when using one of three investigated kinds of wavelet features on the proposed pixel-labeling scheme. This strengthens our previous observations obtained when analyzing visually the results (*cf.* Figures 7(o) and 7(t)). We also observe that the worst performances are mainly obtained when using the Tamura and GLRLM features on the proposed pixel-labeling scheme. The worst overall performances for most of the computed evaluation metrics are noted when using the Tamura features on the proposed pixel-labeling scheme (71%*PPB*). We conclude that the results obtained with the *HBR2013 dataset* strengthen our previous observations with the *DIGIDOC-Texture dataset*.

5.2 Correlation analysis

To highlight the similarities of the behavior of the different evaluated texture features, a correlation analysis of the F-measure performance of each texture-based feature set using the proposed pixel-labeling scheme on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset* is illustrated in Figure 13. The Pearson's linear correlation plots of texture-based feature pairs are represented off diagonal in each figure. The horizontal and vertical axes represent the F-measure performance obtained by using the corresponding texture-based feature sets. Each dot in each correlation plot represents one HDI. The Pearson's linear correlation coefficients highlighted in red indicate which pairs of texture-based feature sets have correlations significantly different from zero.

It can be seen from the different correlation plots in Figure 13 that the Haar-based, the Db3-based and the Db4-based feature sets are highly correlated since they are three wavelet-based approaches and they share similar properties in the localization of the spatial frequency and multi-resolution. Also, the Gabor-based and the wavelet-based features show strong correlation since they both are based on the analysis of the frequency characteristics. The auto-correlation and the wavelet features are also highly correlated. This is due to the fact that the auto-correlation and wavelet-based have similar characteristics. The auto-correlation features are mainly used to find periodic and similar patterns by extracting significant orientations in a texture while the wavelet features are based on a multi-resolution analysis to characterize spatial frequency in a texture. We observe variable values of the computed Pearson's linear correlation coefficients in the correlation plots between the GLCM features and the other analyzed features on the one hand, and between the GLRLM features and the other analyzed features on the other hand. This is due to the fact that the GLRLM and the GLCM features are both based on analyzing the gray-level values by computing whether the probability of a specific run-length or the probability of occurrence of pixel pairs. However, in the case of HDIs with a large amount of noise, the GLRLM and the GLCM features fail to characterize properly a texture since they are sensitive to noise.

An interesting conclusion that can be deduced from the correlation plots in Figure 13 is that the texture-based feature sets perform very well on somes HDIs, while failing on some other HDIs. This confirms our observation when analyzing the F-measure standard deviations of the nine investigated texture-based feature sets in this study using the proposed pixel-labeling scheme on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset* (*cf.* Table 12). It is worth noting that combining the pixel-labeling results of different kinds of texture features can be significantly more relevant.

5.3 Statistical analysis

Demšar [23] recommended the Friedman's test for comparison of more classifiers over multiple datasets because it is considered a simple, safe, robust and non-parametric test. Therefore, we propose in this study to represent the statistical significance of the obtained performance with a graph of the estimates and the comparison intervals of the F-measure performance of the nine investigated texture-based feature sets using the Friedman's test in order to draw one general conclusion from the performance of the analyzed texture features over the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset*. The Friedman's test is based on ranking the different analyzed algorithms for each dataset separately. Figure 6 illustrates a graph of the estimates and the comparison intervals of the F-measure performances of the nine investigated texture-based feature sets using the Friedman's test.

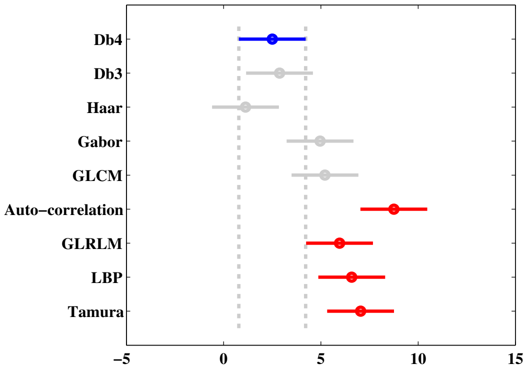


Fig. 6: Graph of the estimates and the comparison intervals of the F-measure performance of the nine investigated texture-based feature sets using the Friedman's test.

In Figure 6, each F-measure mean of texture-based feature set is marked by a circle, and the interval is represented by a horizontal bar extending out from the circle. Two F-measure means of two different texture-based feature sets are significantly different if their intervals are disjoint, while they are significantly similar if their intervals overlap. For instance, if the Db4 bar is selected (the blue bar in Figure 6), we observe that the auto-correlation, the GLRLM, the LBP and the Tamura features have mean column ranks significantly different from the Db4 features (the red bars in Figure 6). Therefore, the bars for the Db3, the Haar, the Gabor and the GLCM features are not significantly different (the gray bars in Figure 6).

We also observe that the performances of the Haar, the Db3 and the Haar features are significantly similar. This confirms our previous observations that they are three wavelet-based approaches and they share similar properties in the localization of the spatial frequency and multi-resolution. This experiment again seem to favor the idea to combine the output of more than one kind of texture feature over the separate analysis of texture-based feature set to in order to improve the pixel-labeling performance.

5.4 Observations and recommendations

Based on the experimental results and observations presented in the previous sections, the effectiveness of a texture-based approach has been shown in HDIA. Hence, a texture-based approach is applicable to a large variety of HDIs. This confirms our initial hypotheses that different document contents have distinct texture features. We observe that specific sets of texture features are more adequate than other ones when the analyzed HDIs have particular contents. Therefore, some observations and recommendations have been deduced about the choice of the used texture feature set according to the particularities of the analyzed HDIs. These observations and recommendations are based on analyzing texture features without formulating a hypothesis concerning the HDI layout (e.g. column layout) or its content (e.g. font size and type) and respecting a constructive compromise between the pixel-labeling quality (*cf.* Section 5.1.2), the performance evaluation (*cf.* Section 5.1.3) and the computational cost (*cf.* Table 13).

- The best performing kind of texture features is the Gabor ones for all types of HDI content. The Gabor-based approach yields a better output than the eight other extracted texture features for almost all computed evaluation accuracy metrics without taking into consideration the spatial relationships of pixels. Nevertheless, the feature dimension of the Gabor-based approach is relatively high. This requires a relatively higher computing time and a lot of computer memory.
- When the numerical complexity and performance evaluation are taken into account by comparing the two best investigated texture-based approaches (*i.e.* the Gabor and wavelet-based approaches), the Gabor one would be the better choice for segmenting different content types, without formulating a hypothesis concerning the HDI layout or its content. Nevertheless, we observe that the Gabor features have more difficulty separating textual regions when they are too spatially close to the graphical ones.
- The two kinds of wavelet features, Db3 and Db4, perform better than the Haar one for all kinds of HDI content. The counterpart for the robustness of using the Db4 and Db3 features is a higher computing time.
- The computational cost of using the auto-correlation and LBP features is similar. However, the auto-correlation-based approach performs considerably better than the LBP one when comparing their pixel-labeling quality and computed accuracy metrics.
- In the case of HDIs with a large amount of noise, the GLRLM and the GLCM features are both not appropriate. This is due to the fact that are based on analyzing the gray-level values by computing whether the probability of a specific run-length or the probability of occurrence of pixel pairs.

- For HDIs containing only text, the performances of the Tamura, the LBP and the GLRLM features are less satisfactory, compared to the other investigated texture-based approaches even if the numerical complexity is sufficiently adequate.
- For segmenting HDIs containing only textual regions with distinct fonts, the Gabor-based approach performs considerably better.
- For distinguishing textual regions from graphical ones, the wavelet-based approach is more suitable. However, when the numerical complexity is taken into account, the wavelet-based approach is the highest resource-consuming one.
- For HDIs containing graphics and single text font, the GLCM features should be a good choice as it is fast and easy to use. Indeed, the lowest time is required to process a page. Nevertheless, the GLCM features are not adequate for separating different text fonts even when it is the less time-consuming.
- For HDIs containing graphics and text, the auto-correlation approach is an effective and efficient texture-based one.
- When the HDI under consideration containing graphics and text than only text, the auto-correlation and the GLCM features perform considerably better.

It is worth noting that there is awareness that many factors (e.g. binary HDIs, low resolution digitization, defined ground truth, number of classes defined in the ground truth, used pixel-labeling scheme for comparing texture, type of used pre-processing stage, kind of used feature extraction technique) can influence the comparative study and experimental evaluation. Therefore, we evaluate a number of commonly and widely used texture features in this article. In this study, we aim at analyzing properly texture features by raising issues related only to how these texture-based sets are compared with each other. We have planned to avoid all unnecessary biases caused by introducing a feature selection task (e.g. dimension reduction technique) or by integrating a post-processing phase based on the analysis of the topological or spatial relationships (e.g. hierarchy, inclusion or neighborhood position). Indeed, it is highly probable that if we introduce a post-processing task, a significant performance improvement can be noted for a given corpus. Nevertheless, this positive performance improvement can not be ensured if another HDI corpus will be assessed. As a consequence, based on a review of the literature we have made a first reasonable attempt as much as possible to carry out a properly and appropriate comparative study on HDIs by using a standard pixel-labeling scheme for evaluating and benchmarking texture features. We are interested in determining which texture methods are firstly well suited for segmenting graphical regions from textual ones, discriminating text in a variety of situations of different fonts and scales and secondly in finding a constructive compromise between the performance and the computational cost.

6 Conclusions and further work

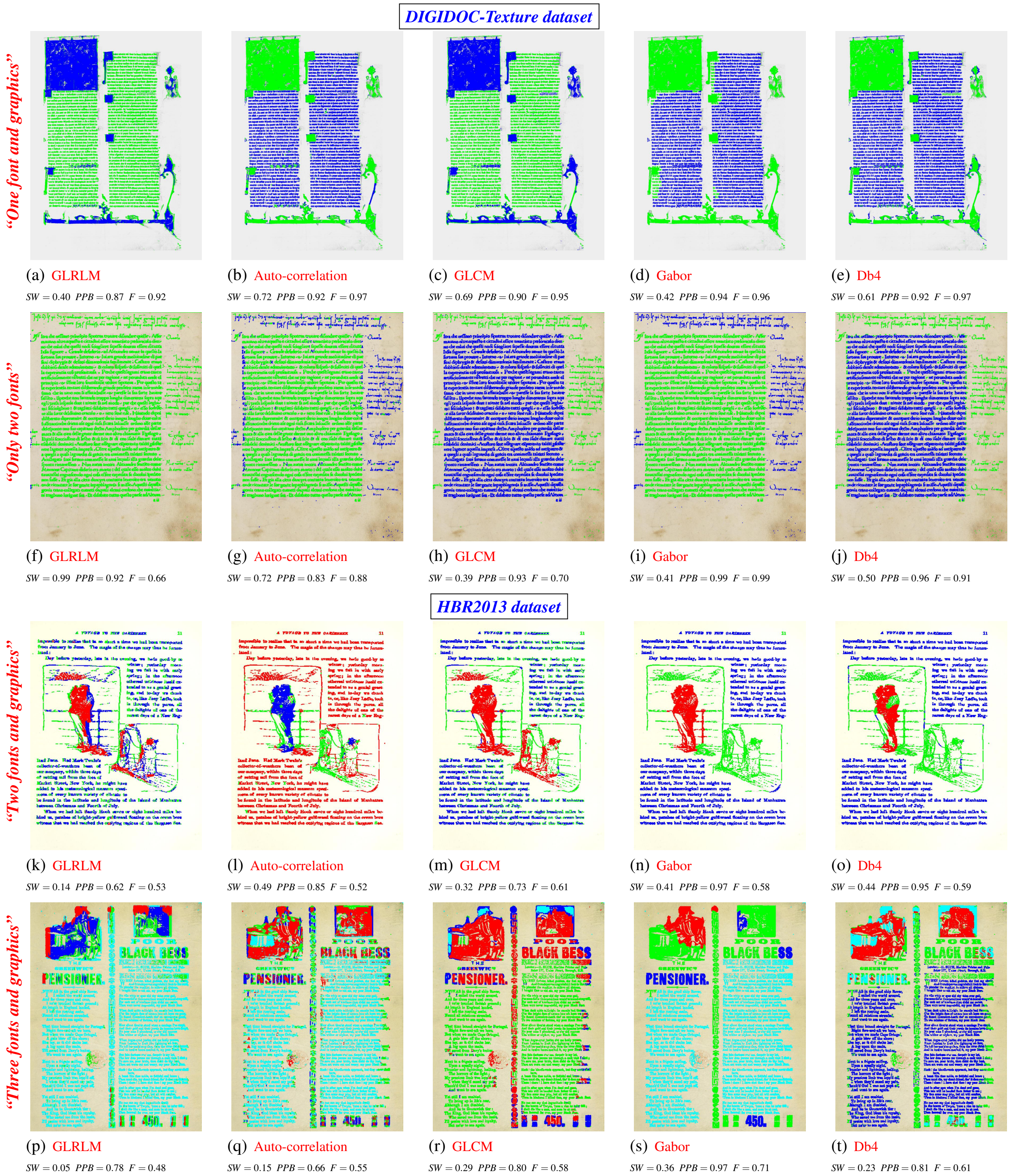
This article has presented an experimental evaluation and benchmarking of a number of commonly and widely used texture features. This comparative study has been conducted on a large corpus of HDIs for the purpose of determining the performance of each texture-based feature set according to the DI content (*i.e.* segmenting graphical regions from textual ones on the one hand, and discriminating text in a variety of situations of different fonts and scales on the other hand). The experimental corpus is composed of two datasets, the *DIGIDOC-Texture dataset* and *HBR2013 dataset*. We have shown the scalability of nine evaluated texture-based feature sets for both datasets. Thus, a standard pixel-labeling scheme for evaluating and benchmarking texture features has been proposed in this study to compare nine texture-based feature sets.

This study has shown the effectiveness of the texture analysis approaches for HDI characterization. Based on our experiments and observations, a thorough analysis of the strengths and the weaknesses of nine well-known texture-based feature sets for HDIA is presented. We conclude that the Gabor and Db4 wavelet features are the best choices for discriminating textual contents from graphical ones without taking into account the spatial relationships between pixels. However, when the numerical complexity and pixel-labeling performance are taken into account, the Gabor approach would be the better choice. Furthermore, the Gabor approach is a good choice for segmenting HDIs containing only textual regions with different fonts. The results reported in this study provide a useful benchmark in terms of performance evaluation, texture vector dimensionality, memory requirements, processing time and complexity for current and future research efforts in HDIA.

There are several possibilities that stem from this study. In particular, improvements can be made regarding the computational cost by introducing an optimization process by means of the single instruction, multiple data parallelization on different general-purpose processing on graphics processing units (GPGPU) to significantly reduce the complexity and the time consumption of the nine investigated texture feature sets. Besides, HDIA is still an open issue for both supervised and unsupervised methods due to the variability of the contents and/or layouts of historical documents. As for the supervised methods, feature learning or representation learning [7] will be investigated for pixel-classification in future research. This helps dealing with retrieving relevant features or representations from raw data. In addition, a feature selection step can also be integrated to select relevant features and remove redundant ones.

Acknowledgment

This study was supported by the French national research agency (ANR), under Grant ANR-10-CORD-0020, which is gratefully acknowledged. The authors would like also to thank Geneviève Cron and Christos Papadopoulos for providing access to the Gallica digital library[1] and IMPACT dataset[2], respectively.



(a) GLRLM $SW = 0.40$ $PPB = 0.87$ $F = 0.92$

(b) Auto-correlation $SW = 0.72$ $PPB = 0.92$ $F = 0.97$

(c) GLCM $SW = 0.69$ $PPB = 0.90$ $F = 0.95$

(d) Gabor $SW = 0.42$ $PPB = 0.94$ $F = 0.96$

(e) Db4 $SW = 0.61$ $PPB = 0.92$ $F = 0.97$

(f) GLRLM $SW = 0.99$ $PPB = 0.92$ $F = 0.66$

(g) Auto-correlation $SW = 0.72$ $PPB = 0.83$ $F = 0.88$

(h) GLCM $SW = 0.39$ $PPB = 0.93$ $F = 0.70$

(i) Gabor $SW = 0.41$ $PPB = 0.99$ $F = 0.99$

(j) Db4 $SW = 0.50$ $PPB = 0.96$ $F = 0.91$

(k) GLRLM $SW = 0.14$ $PPB = 0.62$ $F = 0.53$

(l) Auto-correlation $SW = 0.49$ $PPB = 0.85$ $F = 0.52$

(m) GLCM $SW = 0.32$ $PPB = 0.73$ $F = 0.61$

(n) Gabor $SW = 0.41$ $PPB = 0.97$ $F = 0.58$

(o) Db4 $SW = 0.44$ $PPB = 0.95$ $F = 0.59$

(p) GLRLM $SW = 0.05$ $PPB = 0.78$ $F = 0.48$

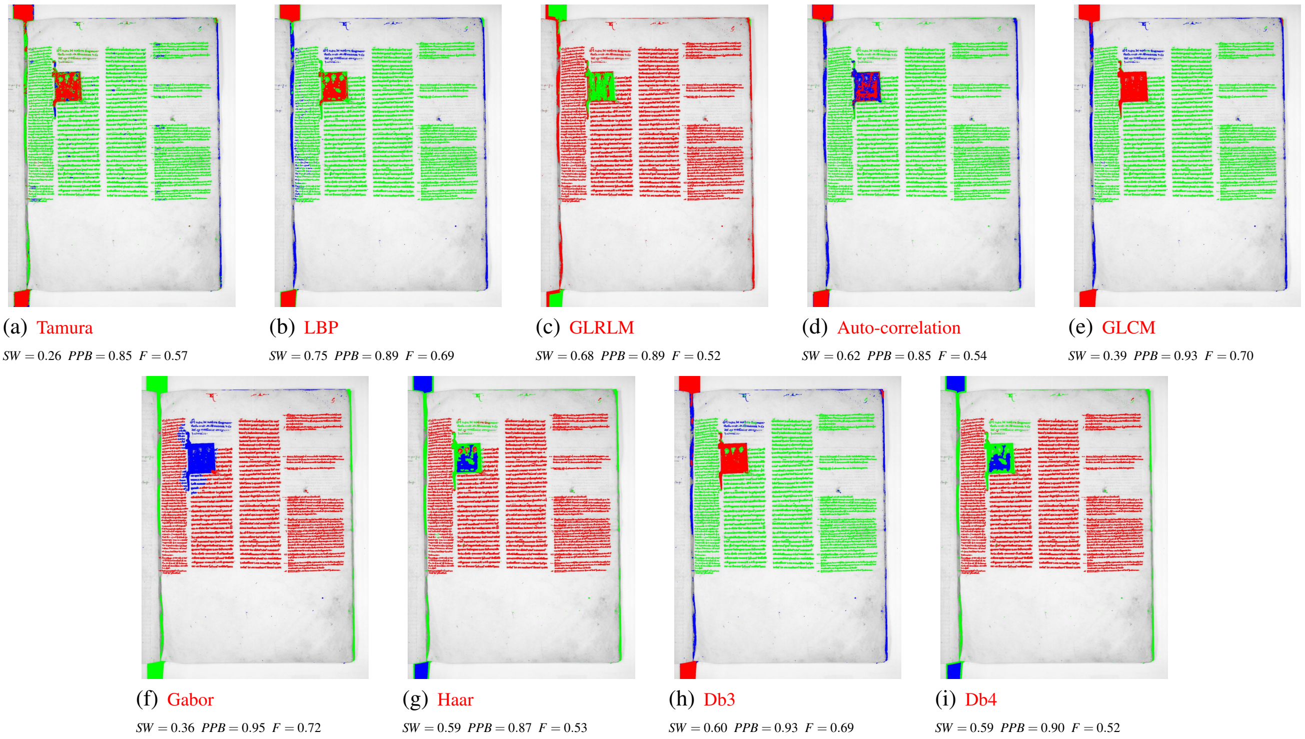
(q) Auto-correlation $SW = 0.15$ $PPB = 0.66$ $F = 0.55$

(r) GLCM $SW = 0.29$ $PPB = 0.80$ $F = 0.58$

(s) Gabor $SW = 0.36$ $PPB = 0.97$ $F = 0.71$

(t) Db4 $SW = 0.23$ $PPB = 0.81$ $F = 0.61$

Fig. 7: Examples of resulting images of the proposed pixel-labeling scheme on the *"One font and graphics"* and *"Only two fonts"* categories of HDIs from the *DIGIDOC-Texture dataset*, and the *"Two fonts and graphics"* and *"Three fonts and graphics"* categories of HDIs from the *HBR2013 dataset*.



(a) Tamura
$SW = 0.26$ $PPB = 0.85$ $F = 0.57$

(b) LBP
$SW = 0.75$ $PPB = 0.89$ $F = 0.69$

(c) GLRLM
$SW = 0.68$ $PPB = 0.89$ $F = 0.52$

(d) Auto-correlation
$SW = 0.62$ $PPB = 0.85$ $F = 0.54$

(e) GLCM
$SW = 0.39$ $PPB = 0.93$ $F = 0.70$

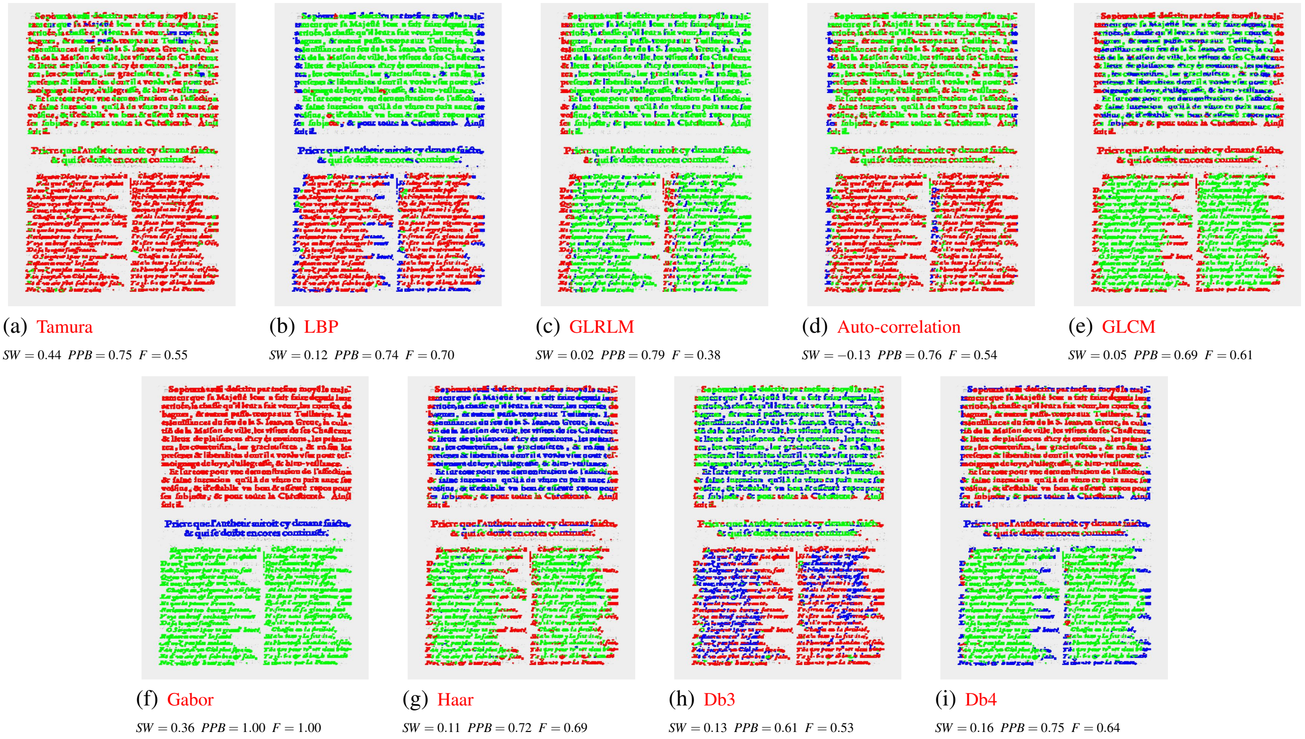
(f) Gabor
$SW = 0.36$ $PPB = 0.95$ $F = 0.72$

(g) Haar
$SW = 0.59$ $PPB = 0.87$ $F = 0.53$

(h) Db3
$SW = 0.60$ $PPB = 0.93$ $F = 0.69$

(i) Db4
$SW = 0.59$ $PPB = 0.90$ $F = 0.52$

Fig. 8: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"Two fonts and graphics"* category of HDIs from the *DIGIDOC-Texture dataset*.



(a) Tamura
$SW = 0.44$ $PPB = 0.75$ $F = 0.55$

(b) LBP
$SW = 0.12$ $PPB = 0.74$ $F = 0.70$

(c) GLRLM
$SW = 0.02$ $PPB = 0.79$ $F = 0.38$

(d) Auto-correlation
$SW = -0.13$ $PPB = 0.76$ $F = 0.54$

(e) GLCM
$SW = 0.05$ $PPB = 0.69$ $F = 0.61$

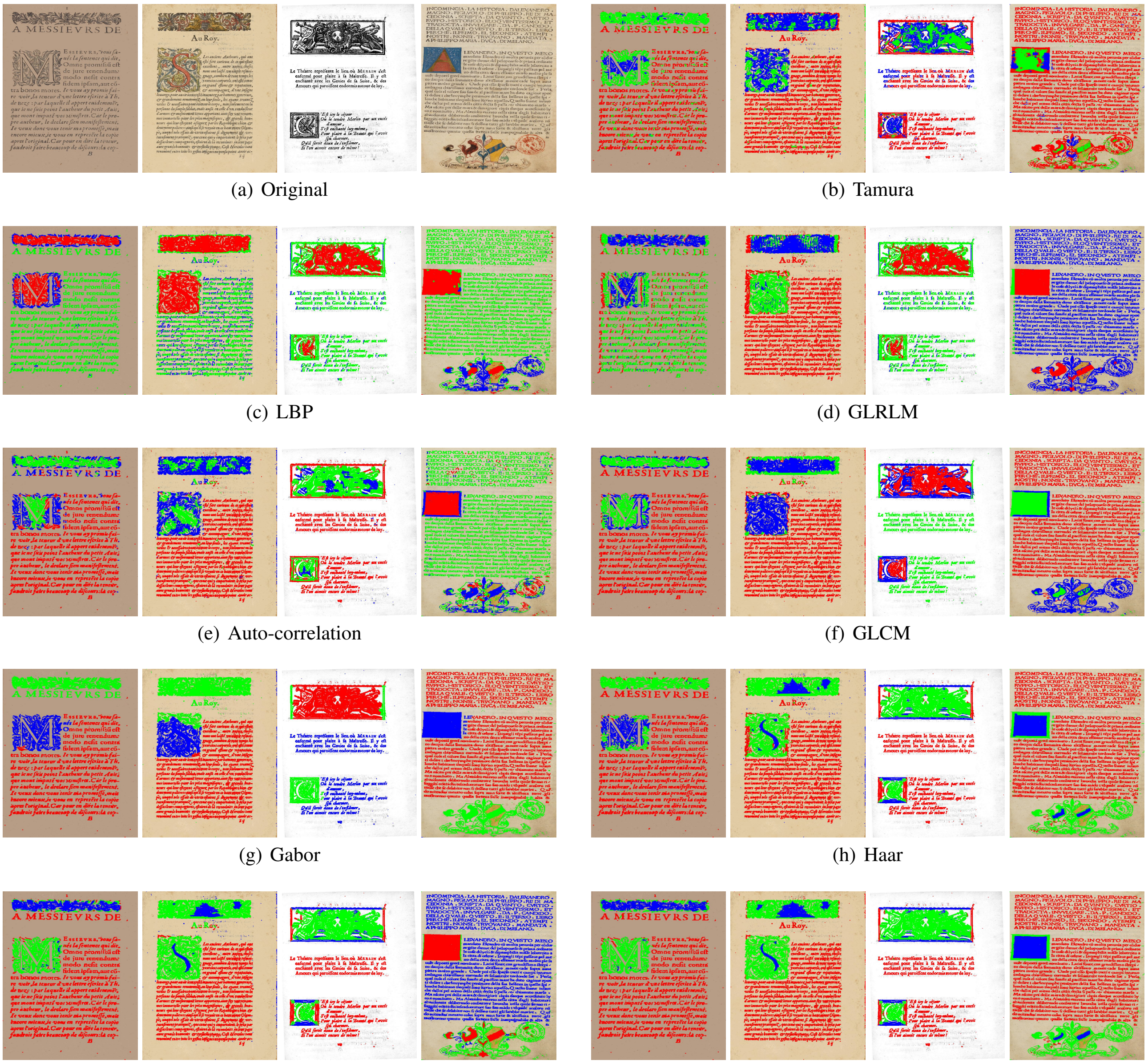
(f) Gabor
$SW = 0.36$ $PPB = 1.00$ $F = 1.00$

(g) Haar
$SW = 0.11$ $PPB = 0.72$ $F = 0.69$

(h) Db3
$SW = 0.13$ $PPB = 0.61$ $F = 0.53$

(i) Db4
$SW = 0.16$ $PPB = 0.75$ $F = 0.64$

Fig. 9: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"Only three fonts"* category of HDIs from the *DIGIDOC-Texture dataset*.



(a) Original

(b) Tamura

(c) LBP

(d) GLRLM

(e) Auto-correlation

(f) GLCM

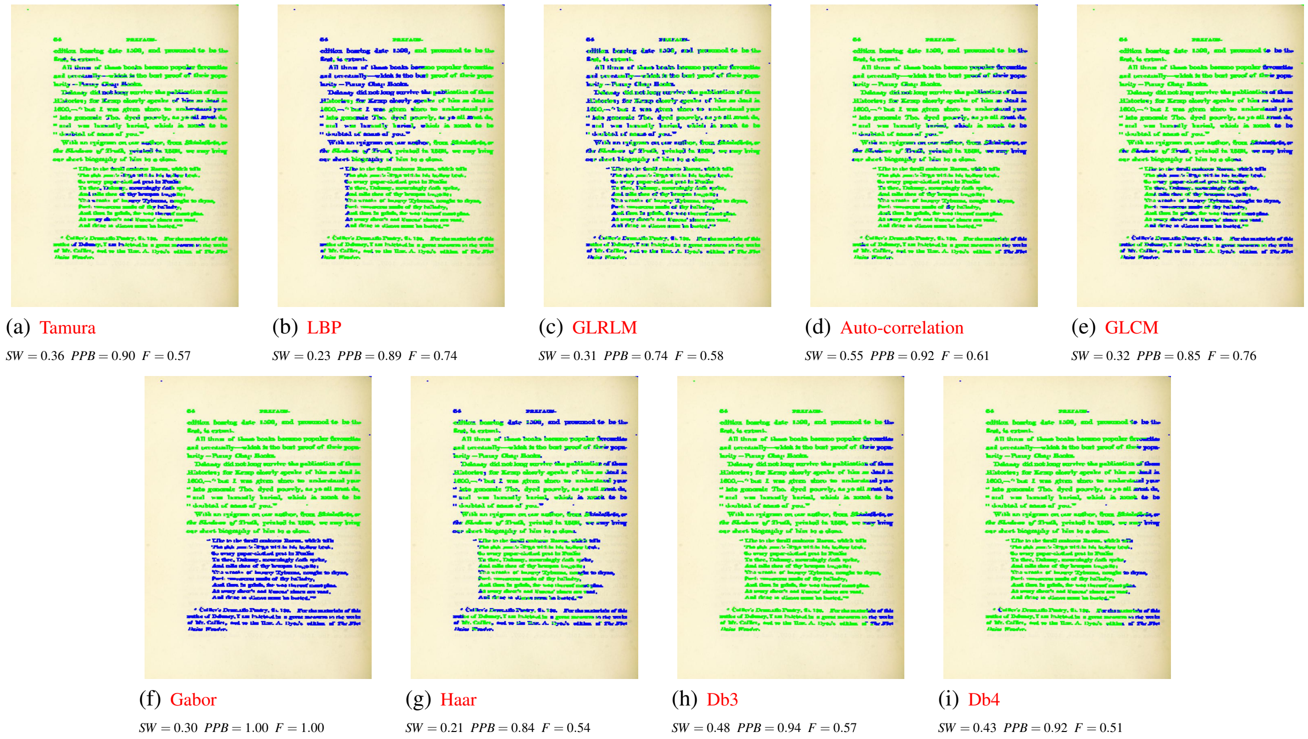
(g) Gabor

(h) Haar

(i) Db3

(j) Db4

Fig. 10: Examples of resulting images of the proposed pixel-labeling scheme on the *"Two fonts and graphics"* category of HDIs from the *DIGIDOC-Texture dataset*.



(a) Tamura
$SW = 0.36$ $PPB = 0.90$ $F = 0.57$

(b) LBP
$SW = 0.23$ $PPB = 0.89$ $F = 0.74$

(c) GLRLM
$SW = 0.31$ $PPB = 0.74$ $F = 0.58$

(d) Auto-correlation
$SW = 0.55$ $PPB = 0.92$ $F = 0.61$

(e) GLCM
$SW = 0.32$ $PPB = 0.85$ $F = 0.76$

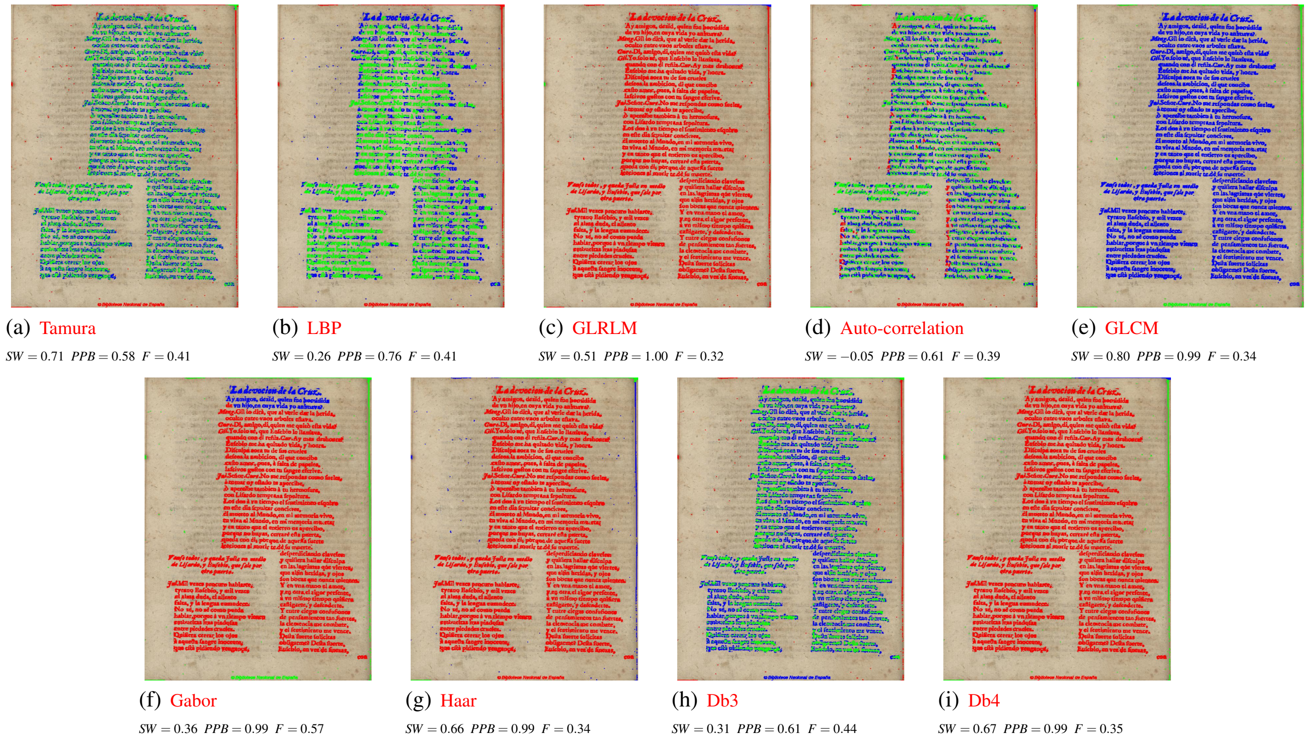
(f) Gabor
$SW = 0.30$ $PPB = 1.00$ $F = 1.00$

(g) Haar
$SW = 0.21$ $PPB = 0.84$ $F = 0.54$

(h) Db3
$SW = 0.48$ $PPB = 0.94$ $F = 0.57$

(i) Db4
$SW = 0.43$ $PPB = 0.92$ $F = 0.51$

Fig. 11: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"Only two fonts"* category of HDIs from the *HBR2013 dataset*.



(a) Tamura
$SW = 0.71$ $PPB = 0.58$ $F = 0.41$

(b) LBP
$SW = 0.26$ $PPB = 0.76$ $F = 0.41$

(c) GLRLM
$SW = 0.51$ $PPB = 1.00$ $F = 0.32$

(d) Auto-correlation
$SW = -0.05$ $PPB = 0.61$ $F = 0.39$

(e) GLCM
$SW = 0.80$ $PPB = 0.99$ $F = 0.34$

(f) Gabor
$SW = 0.36$ $PPB = 0.99$ $F = 0.57$

(g) Haar
$SW = 0.66$ $PPB = 0.99$ $F = 0.34$

(h) Db3
$SW = 0.31$ $PPB = 0.61$ $F = 0.44$

(i) Db4
$SW = 0.67$ $PPB = 0.99$ $F = 0.35$

Fig. 12: Examples of resulting images of the proposed pixel-labeling scheme for comparing the nine investigated texture-based feature sets on the *"Only three fonts"* category of HDIs from the *HBR2013 dataset*.

Table 11: Performance evaluation and comparison of the nine investigated texture-based feature sets in this study using the proposed pixel-labeling scheme on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset*. Internal and external accuracy measures are computed, silhouette width (SW), purity per-block (PPB) and F-measure (F). The higher the values, the better the results. The values which are quoted in **red** and **green** colors, are considered as the **lowest** and **highest**, respectively.

		Tamura	LBP	GLRLM	Auto-correlation	GLCM	Gabor	Haar	Db3	Db4
DIGIDOC-Texture dataset										
One font and graphics	SW	**0.39**	0.57	**0.70**	0.54	0.46	0.51	0.56	0.58	0.57
	PPB	**0.90**	0.91	0.92	0.91	0.92	**0.96**	0.95	0.95	0.95
	F	0.74	0.74	**0.64**	0.82	0.78	**0.88**	0.82	0.84	0.84
Two fonts and graphics	SW	**0.16**	0.37	0.28	0.27	0.30	**0.43**	0.38	0.41	0.40
	PPB	0.86	0.83	**0.80**	0.83	0.86	**0.93**	0.89	0.88	0.91
	F	0.55	0.60	**0.52**	0.59	0.61	**0.67**	0.63	0.62	0.63
Only two fonts	SW	**0.23**	0.37	**0.66**	0.27	0.25	0.39	0.30	0.31	0.30
	PPB	0.87	0.85	0.87	0.84	**0.82**	**0.94**	0.88	0.85	0.88
	F	0.59	0.69	**0.55**	0.72	0.70	**0.84**	0.74	0.73	0.76
Only three fonts	SW	0.16	0.19	0.14	**0.07**	0.15	**0.31**	0.18	0.22	0.19
	PPB	0.84	0.74	**0.70**	0.77	0.74	**0.88**	0.78	0.76	0.79
	F	0.43	0.54	**0.41**	0.61	0.60	**0.64**	0.56	0.57	0.59
Overall	SW	**0.24**	**0.38**	**0.45**	**0.29**	**0.29**	**0.41**	**0.36**	**0.38**	**0.37**
	PPB	**0.87**	**0.83**	**0.82**	**0.84**	**0.84**	**0.93**	**0.88**	**0.86**	**0.88**
	F	**0.58**	**0.64**	**0.53**	**0.69**	**0.67**	**0.76**	**0.69**	**0.69**	**0.71**
HBR2013 dataset										
Only two fonts	SW	0.43	0.51	**0.70**	0.53	0.46	**0.39**	0.50	0.56	0.52
	PPB	**0.90**	0.92	**0.96**	0.95	0.92	0.94	0.91	0.92	0.92
	F	0.54	0.55	**0.50**	0.53	0.56	**0.59**	0.56	0.57	0.56
Two fonts and graphics	SW	0.28	0.36	**0.43**	0.30	**0.18**	0.34	0.39	0.37	**0.43**
	PPB	**0.81**	0.91	0.91	0.84	0.82	**0.92**	0.88	0.88	0.90
	F	0.50	0.43	**0.42**	0.53	0.54	0.55	0.60	0.58	**0.62**
Only three fonts	SW	**0.45**	0.26	0.35	**0.05**	0.33	0.21	0.24	0.29	0.24
	PPB	**0.78**	0.83	**0.87**	0.80	**0.87**	**0.87**	0.83	0.83	0.83
	F	0.39	0.41	**0.37**	0.42	0.40	**0.53**	0.43	0.42	0.43
Three fonts and graphics	SW	0.11	0.31	**0.03**	0.21	0.30	0.39	0.37	**0.43**	0.38
	PPB	**0.70**	0.81	0.80	0.82	0.84	**0.95**	0.86	0.89	0.91
	F	**0.37**	0.40	**0.37**	0.41	0.44	**0.53**	0.41	0.41	0.42
Only four fonts	SW	**0.34**	0.23	**-0.003**	0.05	0.23	0.25	0.18	0.24	0.18
	PPB	0.77	0.76	0.76	**0.72**	0.85	**0.90**	0.73	0.74	0.75
	F	0.32	0.32	**0.29**	0.41	0.34	**0.43**	0.37	0.39	0.37
Four fonts and graphics	SW	**0.32**	0.09	0.04	**-0.07**	0.14	0.26	0.21	0.26	0.21
	PPB	**0.64**	0.77	0.76	0.67	0.81	**0.90**	0.79	0.78	0.79
	F	0.33	**0.29**	**0.29**	0.37	0.38	**0.44**	0.41	0.41	0.43
Only five fonts	SW	**0.53**	0.23	**-0.18**	-0.09	0.27	0.19	0.34	0.27	0.34
	PPB	**0.52**	0.69	0.77	0.59	0.86	**0.89**	0.85	0.73	0.87
	F	0.30	0.24	**0.22**	0.30	0.31	**0.45**	0.29	0.32	0.30
Five fonts and graphics	SW	**0.29**	0.07	-0.08	**-0.15**	0.0008	0.23	0.06	0.13	0.09
	PPB	0.59	0.64	**0.56**	0.61	0.75	**0.88**	0.68	0.66	0.70
	F	**0.29**	0.31	**0.29**	0.39	0.39	**0.55**	0.41	0.44	0.44
Overall	SW	**0.34**	**0.26**	**0.16**	**0.10**	**0.24**	**0.28**	**0.29**	**0.32**	**0.30**
	PPB	**0.71**	**0.79**	**0.80**	**0.75**	**0.84**	**0.91**	**0.82**	**0.80**	**0.83**
	F	**0.38**	**0.37**	**0.34**	**0.42**	**0.42**	**0.51**	**0.43**	**0.44**	**0.45**

Table 12: The evaluation results of the nine investigated texture-based feature sets in this study using the proposed pixel-labeling scheme on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset*, in terms of the F-measure standard deviations.

	Tamura	LBP	GLRLM	Auto-correlation	GLCM	Gabor	Haar	Db3	Db4
DIGIDOC-Texture dataset									
One font and graphics	0.18	0.18	0.16	0.16	0.19	0.17	0.19	0.18	0.19
Two fonts and graphics	0.11	0.14	0.08	0.15	0.15	0.18	0.14	0.14	0.14
Only two fonts	0.11	0.12	0.10	0.11	0.12	0.14	0.13	0.13	0.14
Only three fonts	0.09	0.12	0.06	0.12	0.14	0.17	0.11	0.13	0.13
HBR2013 dataset									
Only two fonts	0.05	0.09	0.04	0.05	0.08	0.13	0.07	0.07	0.07
Two fonts and graphics	0.13	0.13	0.11	0.20	0.15	0.11	0.13	0.15	0.19
Only three fonts	0.03	0.08	0.06	0.07	0.08	0.11	0.07	0.06	0.08
Three fonts and graphics	0.10	0.14	0.12	0.13	0.15	0.21	0.14	0.13	0.16
Only four fonts	0.02	0.07	0.04	0.12	0.10	0.15	0.10	0.11	0.09
Four fonts and graphics	0.06	0.12	0.07	0.05	0.09	0.12	0.09	0.09	0.11
Only five fonts	0.04	0.05	0.06	0.07	0.08	0.13	0.09	0.10	0.11
Five fonts and graphics	0.06	0.11	0.06	0.09	0.07	0.10	0.09	0.09	0.10

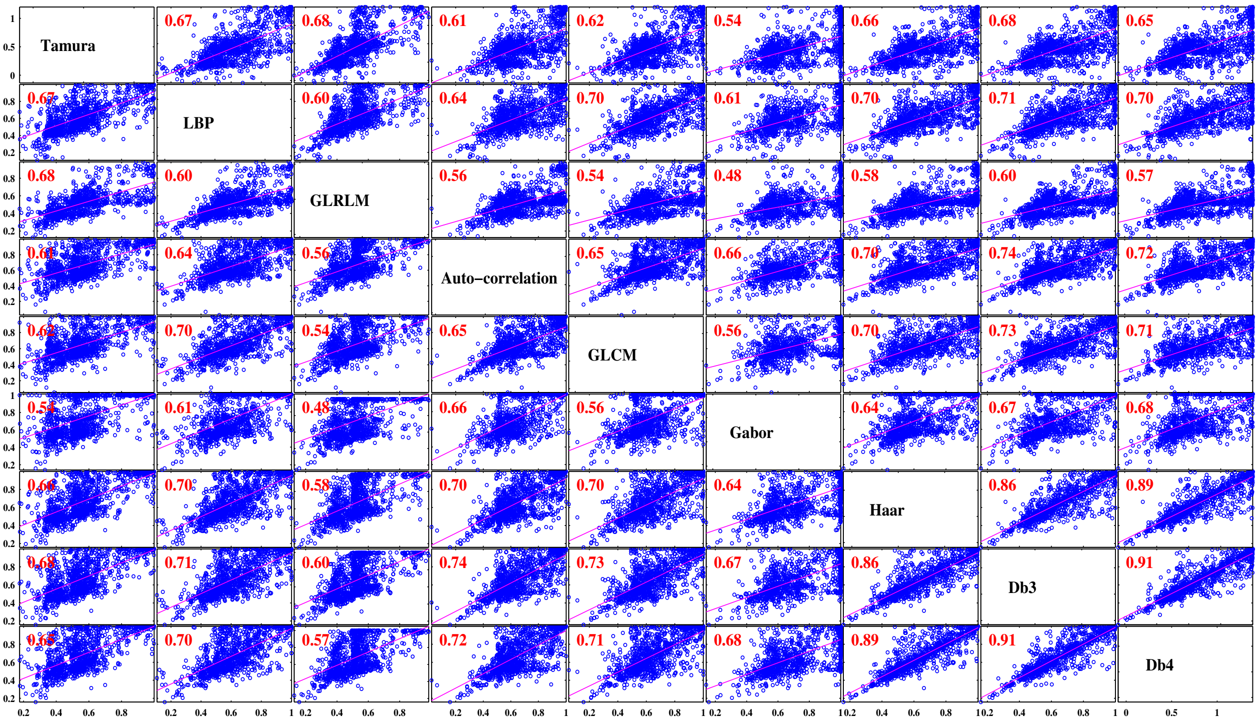


Fig. 13: Correlation analysis of the F-measure performance of each texture-based feature set. The Pearson's linear correlation plots of the F-measure performance of each texture-based feature set using the proposed pixel-labeling scheme on the two datasets, the *DIGIDOC-Texture dataset* and the *HBR2013 dataset* are presented. Each dot in each correlation plot represents one HDI. The Pearson's linear correlation coefficients highlighted in red indicate which pairs of texture-based feature sets have correlations significantly different from zero.

Table 13: Computational cost, benchmarking issues and performance evaluation of the nine investigated texture-based feature sets in this study (*i.e.* memory requirements, processing time, numerical complexity and texture vector dimensionality). An example of the computational cost of extracting and analyzing the nine investigated texture-based feature sets in this study from a full page document scanned at 300 dpi (1965 × 2750 pixels). The values which are quoted in **red** and **green** colors, are considered as the **lowest** and **highest**, respectively.

	Tamura	**LBP**	**GLRLM**	**Auto-correlation**	**GLCM**	**Gabor**	**Haar**	**Db3**	**Db4**
Computational cost									
Texture vector size	$16 = I_t \times N_w$	$40 = I_l \times N_w$	$176 = I_r \times N_w$	$20 = I_a \times N_w$	$72 = I_c \times N_w$	$192 = I_g \times N_w$	$80 = I_h \times N_w$	$80 = I_{db3} \times N_w$	$80 = I_{db4} \times N_w$
Number of texture indices	$I_t = 4$	$I_l = 10$	$I_r = 11\theta_r = 44$	$I_a = 5$	$I_c = 8dc + 2 = 18$	$I_g = 2f_g\theta_g = 48$	$I_h = 20$	$I_{db3} = 20$	$I_{db4} = 20$
Complexity	$O(Mn_t2^{2k})$	$O(MP2^P)$	$O(M\theta_r n_r)$	$O(M(\theta_a N_w \log_2 N_w))$	$O(Md_c n_g^2)$	$O(f_g\theta_g(S^2 \log_2 S))$	$O(M(4JN_w^2 \log_2 N_w))$	$O(M(6JN_w^2 \log_2 N_w))$	$O(M(8JN_w^2 \log_2 N_w))$
Running time	01′14″	02′24″	00′32″	02′33″	00′14″	06′05″	29′17″	37′53″	42′21″
Used memory	≈94 MB	≈53 MB	≈82 MB	**≈48 MB**	**≈587 MB**	≈552 MB	≈61 MB	≈61 MB	≈63 MB
Benchmarking issues and performance evaluation									
Dimensionality	++	++	−	++	+	−	+	+	+
Complexity	+	+	++	+	++	+	−	−−	−−
Used memory	+	+	+	++	−	−	+	+	+
One font and graphics	−	−	−	+	+	++	+	++	++
Two fonts and graphics	−	−	−	+	+	++	+	+	+
Only two fonts	−−	−−	−−	−	−	+	−	−	−
Only three fonts	−−	−−	−−	−	−	+	−	−	−

$I_t, I_l, I_r, I_a, I_c, I_g, I_h, I_{db3}$ and I_{db4} denote the investigated sets of texture features in this article, Tamura, LBP, GLRLM, auto-correlation, GLCM, Gabor, Haar, Db3 and Db4 features, respectively. N_w is the number of sliding windows. In this study, N_w is equal to 4. M is the number of the foreground pixels. $S = W \times H$ is the dimension or size of the input image. W and H denote the effective width and height of the analyzed image, respectively. n_g is the number of gray-levels (*i.e.* 255 gray-levels). n_t is the number of averages $A_{k_t}(x,y)$ for the windows of size $2^{k_t} \times 2^{k_t}$ (*i.e.* 3 averages computed around each selected pixel for the windows of size $2^{k_t} \times 2^{k_t}$, where $k_t = \{0,1,2\}$). P is the number of LBP neighboring pixels (*i.e.* 8 pixels in the neighbor set). θ_r is the number of the angle direction values specified when computing the GLRLM (θ_r=$\{0,\pi/4,\pi/2,3\pi/4\}$). n_r is the number of pixels of the sliding window. θ_a is the possible number of the orientation values of the rose of directions (*i.e.* 179 orientation values). d_c is the GLCM particular distance defined in the probability of the gray-level pairs. In this study, d_c is equal to 2. f_g and θ_g are the spatial frequency and orientation of Gabor filters, respectively (f_g=$\{2\sqrt{2},4\sqrt{2},8\sqrt{2},16\sqrt{2},32\sqrt{2},64\sqrt{2}\}$ and θ_g=$\{0,\pi/4,\pi/2,3\pi/4\}$). In our experiments, the scale of wavelet decomposition J is 3 levels (*i.e.* from first, second and third scale). $I_{A_{2^{-J}}}$, $I_{D^{(v)}_{2^{-j}}}$, $I_{D^{(h)}_{2^{-j}}}$ and $I_{D^{(d)}_{2^{-j}}}$ denote the number of the extracted approximation and detail sub-images features ($1 \leq j \leq J$). $I_h = I_{db3} = I_{db4} = 2I_{A_{2^{-J}}} + 2I_{D^{(v)}_{2^{-1}}} + \cdots + 2I_{D^{(v)}_{2^{-J}}} + 2I_{D^{(h)}_{2^{-1}}} + \cdots + 2I_{D^{(h)}_{2^{-J}}} + 2I_{D^{(d)}_{2^{-1}}} + \cdots + 2I_{D^{(d)}_{2^{-J}}}$

References

1. Antonacopoulos, A., Bridson, D., Papadopoulos, C., Pletschacher, S.: A realistic dataset for performance evaluation of document layout analysis. In: International Conference on Document Analysis and Recognition, pp. 296–300 (2009)
2. Antonacopoulos, A., Clausner, C., Papadopoulos, C., Pletschacher, S.: Historical document layout analysis competition. In: International Conference on Document Analysis and Recognition, pp. 1516–1520 (2011)
3. Antonacopoulos, A., Clausner, C., Papadopoulos, C., Pletschacher, S.: ICDAR 2013 Competition on Historical Book Recognition (HBR 2013). In: International Conference on Document Analysis and Recognition, pp. 1459–1463 (2013)
4. Antonacopoulos, A., Gatos, B., Bridson, D.: Page segmentation competition. In: International Conference on Document Analysis and Recognition, pp. 1279–1283 (2007)
5. Asi, A., Cohen, R., Kedem, K., El-Sana, J., Dinstein, I.: A coarse-to-fine approach for layout analysis of ancient manuscripts. In: International Conference on Frontiers in Handwriting Recognition, pp. 140–145 (2014)
6. Baird, H.S.: Digital libraries and document image analysis. In: International Conference on Document Analysis and Recognition, pp. 2–14 (2003)
7. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. Pattern Analysis and Machine Intelligence pp. 1798–1828 (2013)
8. Beyerer, J., León, F.P., Frese, C.: Texture analysis. In: Machine Vision, pp. 649–683 (2016)
9. Bhowmik, T.K., Kar, M.: Text localization in historical document images with local binary patterns and variance models. In: Lecture Notes in Computer Science - Pattern Recognition and Machine Intelligence, pp. 501–508 (2013)
10. Bovik, A.C., Clark, M., Geisler, W.S.: Multichannel texture analysis using localized spatial filters. Pattern Analysis and Machine Intelligence pp. 55–73 (1990)
11. Busch, A., Boles, W.W., Sridharan, S.: Texture for script identification. Pattern Analysis and Machine Intelligence pp. 1720–1732 (2005)
12. Campbell, F.W., Robson, J.G.: Application of Fourier analysis to the visibility of gratings. The Journal of Physiology pp. 551–566 (1968)
13. Chen, C.H., Pau, L.F., Wang, P.: Texture analysis in the handbook of pattern recognition and computer vision, second edn. World Scientific (1998)
14. Chen, J., Cao, H., Prasad, R., Bhardwaj, A., Natarajan, P.: Gabor features for offline Arabic handwriting recognition. In: International Workshop on Document Analysis Systems, pp. 53–58 (2010)
15. Chen, K., Wei, H., Hennebert, J., Ingold, R., Liwicki, M.: Page segmentation for historical handwritten document images using color and texture features. In: International Conference on Frontiers in Handwriting Recognition, pp. 488–493 (2014)
16. Chen, K., Wei, H., Liwicki, M., Hennebert, J., Ingold, R.: Robust text line segmentation for historical manuscript images using color and texture. In: International Conference on Pattern Recognition, pp. 2978–2983 (2014)
17. Cinque, L., Lombardi, L., Manzini, G.: A multiresolution approach for page segmentation. Pattern Recognition Letters pp. 217–225 (1998)
18. Coppi, D., Grana, C., Cucchiara, R.: Illustrations segmentation in digitized documents using local correlation features. Procedia Computer Science pp. 76–83 (2014)
19. Cote, M., Albu, A.B.: Texture sparseness for pixel classification of business document images. International Journal of Document Analysis and Recognition pp. 1–17 (2014)
20. Coustaty, M., Raveaux, R., Ogier, J.M.: Historical document analysis: a review of French projects and open issues. In: European Signal Processing Conference, pp. 1445–1449 (2011)
21. Cruz-Fernández, F., Ramos-Terrades, O.: Document segmentation using relative location features. In: International Conference on Pattern Recognition, pp. 1562–1565 (2012)
22. Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. Journal of the Optical Society of America A pp. 1160–1169 (1985)
23. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research pp. 1–30 (2006)
24. DuBuf, J.M.H., Kardan, M., Spann, M.: Texture feature performance for image segmentation. Pattern Recognition pp. 291–309 (1990)
25. Eglin, V., Bres, S., Rivero, C.: Hermite and Gabor transforms for noise reduction and handwriting classification in ancient manuscripts. International Journal of Document Analysis and Recognition pp. 101–122 (2007)
26. Ferrer, M.A., Morales, A., Pal, U.: LBP based line-wise script identification. In: International Conference on Document Analysis and Recognition, pp. 369–373 (2013)
27. Gabor, D.: Theory of communication. Part 1: The analysis of information. Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering pp. 429–441 (1946)
28. Galloway, M.M.: Texture analysis using gray level run lengths. Computer Graphics and Image Processing pp. 172–179 (1975)
29. Garz, A., Sablatnig, R.: Multi-scale texture-based text recognition in ancient manuscripts. In: International Conference on Virtual Systems and Multimedia, pp. 336–339 (2010)
30. Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. Pattern Recognition pp. 317–327 (2006)
31. Grana, C., Serra, G., Manfredi, M., Coppi, D., Cucchiara, R.: Layout analysis and content enrichment of digitized books. Multimedia Tools and Applications pp. 1–22 (2014)
32. Haralick, R.M.: Statistical and structural approaches to texture. In Proceedings of the IEEE pp. 786–804 (1979)
33. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. Systems Man and Cybernetics pp. 610–621 (1973)
34. Harwood, D., Ojala, T., Pietikäinen, M., Kelman, S., Davis, L.: Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions. Pattern Recognition Letters pp. 971–987 (1995)
35. He, J., Do, Q.D.M., Downton, A.C., Kim, J.H.: A comparison of binarization methods for historical archive documents. In: International Conference on Document Analysis and Recognition, pp. 538–542 (2005)
36. Hebert, D., Paquet, T., Nicolas, S.: Continuous CRF with multiscale quantization feature functions application to structure extraction in old newspaper. In: International Conference on Document Analysis and Recognition, pp. 493–497 (2011)
37. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: a review. Pattern Analysis and Machine Intelligence pp. 4–37 (2000)
38. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using Gabor filters. Pattern Recognition pp. 1167–1186 (1991)
39. Jain, A.K., Zhong, Y.: Page segmentation using texture analysis. Pattern Recognition pp. 743–770 (1996)
40. Journet, N., Ramel, J., Mullot, R., Eglin, V.: Document image characterization using a multiresolution analysis of the texture: application to old documents. International Journal of Document Analysis and Recognition pp. 9–18 (2008)
41. Keysers, D., Shafait, F., Breuel, T.M.: Document image zone classification - a simple high-performance approach. In: International Conference on Computer Vision Theory and Applications, pp. 44–51 (2007)

42. Kise, K.: Page segmentation techniques in document analysis. In: Handbook of Document Image Processing and Recognition, pp. 135–175 (2014)
43. Kricha, A., Amara, N.E.B.: Exploring textural analysis for historical documents characterization. Journal of computing pp. 24–30 (2011)
44. Kumar, S., Gupta, R., Khanna, N., Chaudhury, S., Joshi, S.D.: Text extraction and document image segmentation using matched wavelets and MRF model. Image Processing pp. 2117–2128 (2007)
45. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning, pp. 282–289 (2001)
46. Lance, G.N., Williams, W.T.: A general theory of classificatory sorting strategies 1. Hierarchical systems. The Computer Journal pp. 373–380 (1967)
47. Li, J., Gray, R.M.: Context-based multiscale classification of document images using wavelet coefficient distributions. Image Processing pp. 1604–1616 (2000)
48. Lin, M., Tapamo, J., Ndovie, B.: A texture-based method for document segmentation and classification. South African Computer Journal pp. 49–56 (2006)
49. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. Pattern Analysis and Machine Intelligence pp. 674–693 (1989)
50. Mehri, M., Gomez-Krämer, P., Héroux, P., Boucher, A., Mullot, R.: Texture feature evaluation for segmentation of historical document images. In: International Workshop on Historical Document Imaging and Processing, pp. 102–109 (2013)
51. Mehri, M., Gomez-Krämer, P., Héroux, P., Boucher, A., Mullot, R.: A texture-based pixel labeling approach for historical books. Pattern Analysis and Applications (2015)
52. Mehri, M., Kieu, V.C., Mhiri, M., Héroux, P., Gomez-Krämer, P., Mahjoub, M.A., Mullot, R.: Robustness assessment of texture features for the segmentation of ancient documents. In: International Workshop on Document Analysis Systems, pp. 293–297 (2014)
53. Mehri, M., Mhiri, M., Héroux, P., Gomez-Krämer, P., Mahjoub, M.A., Mullot, R.: Performance evaluation and benchmarking of six texture-based feature sets for segmenting historical documents. In: International Conference on Pattern Recognition, pp. 2885–2890 (2014)
54. Mikkilineni, A.K., Chiang, P.J., Ali, G.N., Chiu, G.T.C., Allebach, J.P., III, E.J.D.: Printer identification based on graylevel co-occurrence features for security and forensic applications. In: Security, Steganography, and Watermarking of Multimedia Contents VII, pp. 430–440 (2005)
55. Mouats, K., Journet, N., Mullot, R.: Segmentation floue d'images de documents anciens par approche texture utilisant le filtre de Gabor. In: International Conference on Image and Signal Processing (2006)
56. Nguyen, G., Coustaty, M., Ogier, J.M.: Stroke feature extraction for lettrine indexing. In: International Conference on Image Processing Theory Tools and Applications, pp. 355–360 (2010)
57. Nicolaou, A., Slimane, F., Märgner, V., Liwicki, M.: Local binary patterns for Arabic optical font recognition. In: International Workshop on Document Analysis Systems, pp. 76–80 (2014)
58. Nikolaou, N., Makridis, M., Gatos, B., Stamatopoulos, N., Papamarkos, N.: Segmentation of historical machine-printed documents using adaptive run-length smoothing and skeleton segmentation paths. Image and Vision Computing pp. 590–604 (2010)
59. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence pp. 971–987 (2002)
60. Okun, O., Pietikäinen, M.: A survey of texture-based methods for document layout analysis. In: Workshop on Texture Analysis in Machine Vision, pp. 137–148 (1999)
61. Otsu, N.: A threshold selection method from gray-level histograms. Systems, Man, and Cybernetics pp. 62–66 (1979)
62. Petrou, M., Sevilla, P.G.: Image processing: dealing with texture. John Wiley & Sons (2006)
63. Romero, V., Fornés, A., Serrano, N., Sánchez, J.A., Toselli, A.H., Frinken, V., Vidal, E., Lladós, J.: The ESPOSALLES database: an ancient marriage license corpus for off-line handwriting recognition. Pattern Recognition pp. 1658–1669 (2013)
64. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recognition pp. 225–236 (2000)
65. Serrano, N., Castro, F., Juan, A.: The RODRIGO database. In: International Conference on Language Resources and Evaluation, pp. 2709–2712 (2010)
66. Seuret, M., Liwicki, M., Ingold, R.: Pixel level handwritten and printed content discrimination in scanned documents. In: International Conference on Frontiers in Handwriting Recognition, pp. 423–428 (2014)
67. Shafait, F., Keysers, D., Breuel, T.M.: Performance evaluation and benchmarking of six-page segmentation algorithms. Pattern Analysis and Machine Intelligence pp. 941–954 (2008)
68. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. Systems Man and Cybernetics pp. 460–473 (1978)
69. Tang, X.: Texture information in run-length matrices. Image Processing pp. 1602–1609 (1998)
70. Tuceryan, M., Jain, A.K.: Texture segmentation using Voronoi polygons. Pattern Analysis and Machine Intelligence pp. 211–216 (1990)
71. Uttama, S., Loonis, P., Delalandre, M., Ogier, J.M.: Segmentation and retrieval of ancient graphic documents. In: International Workshop on Graphics Recognition, pp. 88–98 (2006)
72. Villegas, M., Romero, V., Sánchez, J.A.: On the modification of binarization algorithms to retain grayscale information for handwritten text recognition. In: Iberian Conference on Pattern Recognition and Image Analysis, pp. 208–215 (2015)
73. Wang, D., Srihari, S.N.: Page segmentation and classification. Computer Vision, Graphics, and Image Processing pp. 327–352 (1989)
74. Wang, L., He, D.C.: Texture classification using texture spectrum. Pattern Recognition pp. 905–910 (1990)
75. Wechsler, H.: Texture analysis - a survey. Signal Processing pp. 271–282 (1980)
76. Weszka, J.S., Dyer, C.R., Rosenfeld, A.: A comparative study of texture measures for terrain classification. Systems Man and Cybernetics pp. 269–285 (1976)
77. Zhu, Y., Tan, T., Wang, Y.: Font recognition based on global texture analysis. Pattern Analysis and Machine Intelligence pp. 1192–1200 (2001)
78. Tuceryan, M.: Moment based texture segmentation. Pattern Recognition Letters pp. 659–668 (1994)