Pattern Recognition 47 (2014) 1051–1062

Contents lists available at ScienceDirect

Pattern Recognition

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

Distinction between handwritten and machine-printed text based on the bag of visual words model

Konstantinos Zagoris [a,b,*], Ioannis Pratikakis [a], Apostolos Antonacopoulos [b], Basilis Gatos [c], Nikos Papamarkos [a]

[a] Visual Computing Group, Department of Electrical and Computer Engineering Democritus University of Thrace, Xanthi, Greece
[b] Pattern Recognition and Image Analysis (PRImA) Research Lab School of Computing, Science and Engineering, University of Salford, Greater Manchester, UK
[c] Institute of Informatics and Telecommunications, National Center for Scientific Research Demokritos Athens, Greece

ARTICLE INFO

Available online 20 September 2013

Keywords:
Bag of visual words
Local features
Support vector machines
Page layout

ABSTRACT

In a variety of documents, ranging from forms to archive documents and books with annotations, machine printed and handwritten text may coexist in the same document image, raising significant issues within the recognition pipeline. It is, therefore, necessary to separate the two types of text so that it becomes feasible to apply different recognition methodologies to each modality. In this paper, a new approach is proposed which strives towards identifying and separating handwritten from machine printed text using the Bag of Visual Words model (BoVW). Initially, blocks of interest are detected in the document image. For each block, a descriptor is calculated based on the BoVW. The final characterization of the blocks as Handwritten, Machine Printed or Noise is made by a decision scheme which relies upon the combination of binary SVM classifiers. The promising performance of the proposed approach is shown by using a consistent evaluation methodology which couples meaningful measures along with new datasets dedicated to the problem upon consideration.

1. Introduction

Nowadays, one can observe a rapidly growing number of digitization initiatives in libraries and archives, involving a variety of document types. Among several other obstacles, the presence of printed and handwritten text in the same document image gives rise to significant issues since each modality requires different treatment to recognize the corresponding characters [1,2]. Furthermore, the automatic processing of application forms, bank checks, petitions, mail items, etc., makes imperative the distinction between handwritten and machine-printed text.

Previous efforts can be separated into three categories based on the level in which the text classification is performed, that is text lines, words or characters.

In the case of text lines level classification, Pal and Chaudhuri [3,4] present a method which separates the machine-printed from handwritten text lines in Bangla and Devnagari, two popular scripts in south Asia. The authors use a technique based on structural and statistical features of machine-printed and handwritten text lines that appear in these scripts. For the line segmentation, they detect horizontal or vertical text lines using their corresponding projection profiles. For the classification scheme, they use a three-tier tree classifier employing some simple structural features tailored to the two specific scripts under study. Kavallieratou and Stamatatos [5] examine only the horizontal projection of the upper and lower profiles of a detected text line to separate the handwritten part from the machine-printed. Then, they employ discriminant analysis to the extracted feature set in order to classify the lines as handwritten or machined-printed. Santos et al. [6] use a fixed size window on a set of base lines and they extract content and shape related features in order to detect handwritten writing in bank checks.

In the case of word level classification between handwritten and machine-printed text, Guo and Ma [2] segment the image document into word-blocks by detecting connected components that are subsequently merged based on a set of conditions. For each word-block, a projection profile is created which is linearly quantized. The classification of the aforementioned sequence as handwritten or machine-printed text is achieved by using Hidden Markov Models. Da Silva and Conci [7] developed a system that analyses various types of application forms, such as subscription forms, questionnaires or preprinted memorandums. They, initially,

* Corresponding author at: Visual Computing Group, Department of Electrical and Computer Engineering Democritus University of Thrace, Xanthi, Greece. Tel.: +30 2541079577.
E-mail addresses: kzagoris@ee.duth.gr (K. Zagoris), ipratika@ee.duth.gr (I. Pratikakis), a.antonacopoulos@primaresearch.org (A. Antonacopoulos), bgat@iit.demokritos.gr (B. Gatos), papamark@ee.duth.gr (N. Papamarkos).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.09.005

segment the forms into word-blocks using Connected Component Analysis for which, eleven features are extracted. Finally, each word block is classified as handwritten or machined-printed text using pre-determined thresholds based on training data. Farooq et al. [8] employ Gabor filters on word blocks followed by classification using a probabilistic neural network in order to detect handwritten Arabic text. Peng et al. [9] model the entire image document as Markov Random Field and separate it in three different classes (machine printed text, handwritten text and overlapped text). Zheng et al. [10] identify machined printed and handwriting text in noisy document images. They calculate the connected components in a page and then they merge them aiming to form word blocks based on spatial proximity. For the text identification (handwritten, machine-printed or noise) they initially extract several sets of features like Gabor filter, crossing count histogram and Bi-level co-occurrence. For the classification, the Fisher classifier is considered.

In the case of a character level classification, Fan et al. [11] propose a method to initially detect the orientation of a text block by analyzing the valleys in horizontal and vertical projection profiles. Then, the character blocks are obtained by employing an X–Y cut algorithm. Finally, the classification task is addressed using character block histograms that incorporate spatial information.

The application scope of the previous methods is limited to a single context. Particularly, in the case of text lines classification, handwriting annotations cannot be handled. The existing approaches which deal with classification on word level are affected by the failures on the segmentation stage thus, they restrict their applicability to particular document domains like bank checks or forms wherein the layout is predictable. Last but not least, in the case of methods that classification is addressed at the character level it is difficult to deal with noisy content which expands in size larger than the size of a character.

In this paper, we propose a new approach dealing with the problem of handwritten and machine-printed text separation using the Bag of Visual Words (BoVW) model. In contrast to previous approaches, it can identify (and separate from type-written text) also handwritten annotations in addition to complete handwritten paragraphs.

The novelty of the proposed method is based upon the following: (i) the use of BoVW model to separate machine printed and handwritten textual information in document images coupled with an optimal codebook creation using a Self-Growing and Self-Organized Neural Gas (SGONG) network. Moreover, this addresses the main shortcoming of BoVW models which use a fixed number of clusters to build a visual dictionary; (ii) the incorporation of a final classification which takes into account a decision step that relies upon a combination of binary SVM classifiers. Therefore, it augments current classifiers performance by introducing an explicit decision system; (iii) the generic nature of proposed method that deals with document images which originate from datasets that come from different machine-printed/handwritten separation context. This novelty addresses the current weakness of available systems which deal with datasets in specific context; (iv) meaningful performance evaluation addressed by the incorporation of corresponding measures which are suitable to the machine-printed and handwritten separation problems to overcome ambiguities by evaluation measures which are not directly related to the problem at hand.

Additionally, we provide three public available distinct datasets each one containing different machine/handwritten separation context. This removes the existing obstacles in the literature and makes the evaluations of future approaches more easy.

The paper is structured as follows: Section 2 details the proposed methodology, Section 3 discusses the evaluation framework along with the corresponding experimental results and finally, in Section 4, conclusions are drawn.

2. The proposed methodology

2.1. Bag of visual words (BoVW) model

The BoVW model is inspired by the Bag of Words (BoW) model employed in information retrieval in which a document is described by a set of words. Accordingly, the BoVW model for document images comprises a set of "visual words" to describe the image content [12].

A "visual word" is expressed by a set of features that correspond to local image information of the image pixels which is identified by the image keypoints [13]. These features are grouped in a number of clusters. A "visual word" is denoted as the vector which represents the features of each cluster centroid while the set of all clusters defines a codebook which is analogous to a visual dictionary. In particular, each local point belongs to a visual word which corresponds to the closest center of the cluster calculated by a distance function such as Euclidean, Manhattan, etc. Finally, the image is represented by a vector which denotes the corresponding descriptor and it reflects the frequency of each visual word that appears in the image. Fig. 1 illustrates the BoVW paradigm.

In the literature, a number of local features have been presented. The most well-known local features are the Scale-Invariant Feature Transform (SIFT) [14], and the Speeded-up Robust Features (SURF) [15]. These features have been proved useful due to their invariance to scale and rotation as well for the robustness across considerable range of distortion, noise contamination and change in brightness.

The SIFT descriptor is highly discriminant but, being a 128-vector, it is relatively slow to compute and match. Similar to SIFT, SURF relies on local gradient histograms but uses integral images to speed up the computation. Different parameter settings are possible but, since a vector of 64 dimensions already yields good recognition performance, that version has become a de facto standard.

Moreover, another family of local keypoints and descriptors have emerged which are focusing on more efficient and effective calculation. These local features are binary and provide high performance at a dramatically lower computational cost in document image processing applications. The BRIEF [16] and BRISK [17] local descriptors are members of this family.

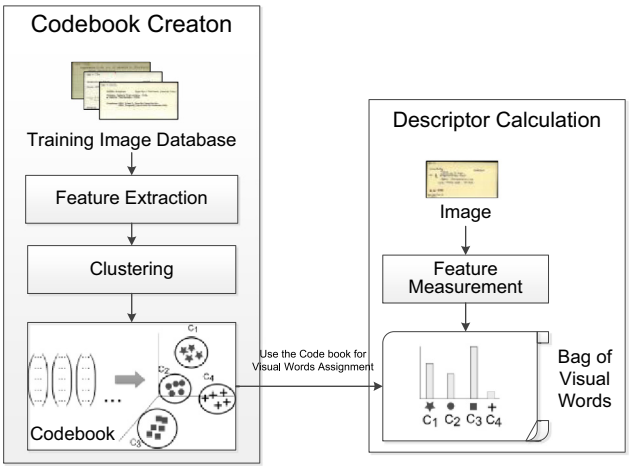

Fig. 1. The BoVW paradigm.

BRIEF uses the SURF keypoints and for the descriptor calculation it uses the Local Binary Patterns (LBP) to translate the keypoint neighborhoods (circles of fixed radius) into its decimal representation and finally, build a concatenation histogram of these values.

In the Binary Robust Invariant Scalable Keypoints (BRISK) points of interest are identified across both the spatial and scale domains of the image using a saliency criterion. The improved efficiency of the keypoints computation stems from their detection in the octave layers of the image pyramid as well as in layers in-between. The location and the scale of each keypoint are obtained in the continuous domain via quadratic function fitting. For the calculation of the descriptor keypoint a sampling pattern consisting of points residing on an appropriately scaled concentric circles is applied at the neighborhood of each keypoint to retrieve intensity values. Finally, the oriented BRISK sampling pattern is used to obtain pair-wise brightness comparison results which are assembled into a binary descriptor.

There has been work based on BoVW in a variety of application areas. Bolovinou et al. [18] introduce a novel bag of spatio-visual words model which can be combined with a standard BoVW model in order to add local context in the representation of the scenes for successful classification. Nilsback and Zisserman [19] introduce a flower classification technique by developing a bag of visual words model. They show that their work surpasses the baseline algorithms. Deselaers et al. [20] present an adult content image detection and filtering method based on the BoVW classification model. They demonstrate that integrating standard skin colour features into their system led to an improvement compared to the standard model.

Recently, works on document image processing have been presented. Rothacker et al. [21] use bag-of-features representations for estimating a semi-continuous HMM for Arabic handwriting recognition. Shekhar and Jawahar [22] retrieve similar word images based on Bag of Visual Words approach for four Indian languages. They use SIFT local descriptors for word image representation.

It is worth noting, however, that to the best of the authors knowledge there is no approach using the BoVW model to discriminate handwritten from machine printed text in document images. The proposed incorporation of this model to the separation of machine printed from handwritten text is illustrated in Fig. 2, which shows the main stages of the proposed method. It is composed of three stages:

1. *Text Block Segmentation*: The objective of this stage is to detect blocks of interest in the document image (Section 2.2).
2. *Block Descriptor Extraction*: In this stage, the descriptor is calculated based on the BoVW model (Section 2.3). Then, a weight is applied based on the statistics of the datasets. Section 2.3.3 describes the various weighting schemes of block descriptor .
3. *Classification*: The final stage in which a classification system decides what text type (if any) resides in the block based on its descriptor set (Section 2.4).

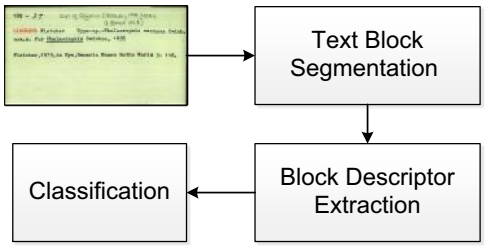

Fig. 2. The main stages of the proposed method.

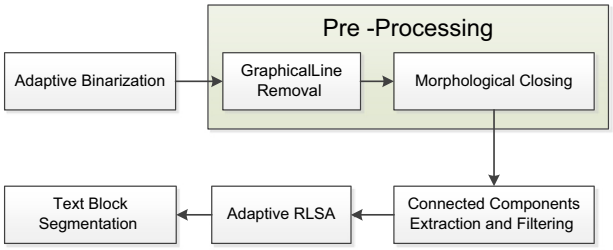

Fig. 3. The steps for text block segmentation.

2.2. Text block segmentation

The main objective of this stage is to detect textual patches in the document image. Fig. 3 shows the consecutive steps of the proposed methodology. During this stage a number of challenges have to be addressed which could be grouped in four categories:

- Binarization problems which stem from different writing instruments (typewriter, pencil, pens of various colours, stamps) and multiple text grey level profiles appearing in the document. Representative examples are shown in Fig. 4a and b.
- Overlapping of handwritten and machine printed text (Fig. 4c and d).
- Overlapping of text with noise as shown in Fig. 4e and f.
- Combinations of the above categories in which overlapping text (handwritten/machined printed) coexist with noise in different strokes.

To deal with the aforementioned challenges, initially a locally adaptive binarization method [23] is applied on the original image (Fig. 5b) which improves the quality of degraded documents enhancing the textual information without requiring any parameter tuning. Afterwards, a Hough transform [24,25] is employed in order to detect and remove graphical straight lines in any orientation in order to avoid large CCs. (Fig. 5c). Then, a morphological closing operation is employed in order to reconstruct the characters shape which might be destroyed by the previous step (Fig. 5d). The aforementioned operations have the advantage of separating overlapping occurrences of background noise with text characters with minimum textual information loss.

Afterwards, the connected components (CCs) of the image are extracted (Fig. 5e) and the noisy elements are filtered out (Fig. 5f) based on the following three criteria: (i) $H(CC) < 5$ or $W(CC) < 5$, (ii) $D(CC) < 0.05$ or $D(CC) > 0.9$, (iii) $E(CC) < 0.08$, where $H(CC)$ and $W(CC)$ denote the bounding box *Height* and *Width* respectively, $E(CC)$ denotes the *Elongation*

$$E(CC) = \frac{\min\{H(CC), W(CC)\}}{\max\{H(CC), W(CC)\}}$$

and $D(CC)$ denotes the *Textual Density*

$$D(CC) = \frac{Fn(CC)}{H(CC) \cdot W(CC)}$$

which is the ratio of the number of foreground pixels $Fn(CC)$ to the total number of pixels in the bounding box.

The above criteria have been chosen heuristically after experimental work, taking into account that CCs should contain text. We assume that CCs with very low width or height and those that have low or high textual (black pixel) density should be considered as noisy non-text objects and thus, proceed to filtering them out. The heuristics used does not affect the selection of textual content with different orientations and font types. In the case of the size of the block, detection will take into account the size limits that have

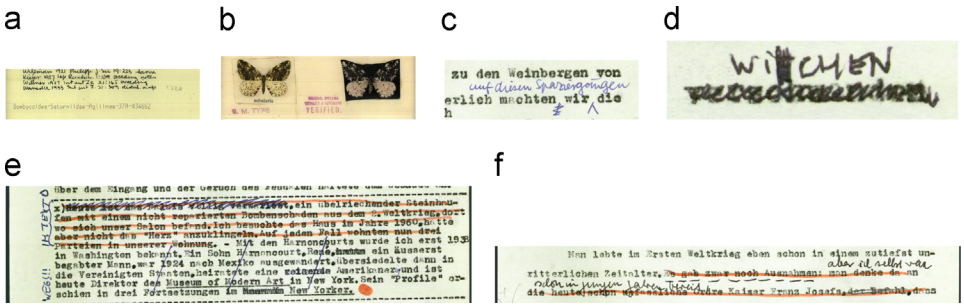

Fig. 4. Block segmentation challenges: (a) and (b) multiple text grey level profiles, (c) and (d) machine printed–handwritten text overlapping, (e) and (f) text–noise overlapping.

Fig. 5. (a) Original image, (b) binarized image, (c) graphical line removal, (d) morphological closing, (e) CCs before filtering, (f) CCs after filtering, (g) ARLSA output, (h) textual component detection, and (i) final results.

been set in the chosen criteria. However, these criteria have been selected in such a way that meaningful textual information is not omitted.

The next step involves merging of CCs in order to build blocks of interest containing textual information. This task is accomplished by the Adaptive Run Length Smoothing Algorithm (ARLSA) [26] (Fig. 5g), which is a modified version of the horizontal RLSA. This method addresses challenges like text with various font sizes, high proximity of text and non-text areas as well as overlapping text lines.

The next task is to split the outcome of the ARLSA (mostly text lines) into smaller textual components. The aim is to identify uniform components that on one hand should be large enough to contain the necessary information for good discrimination performance and on the other hand should be small enough not to create ambiguities.

Towards that goal, first the vertical projection of the lines (Fig. 7b) and afterwards the histogram of the consecutive zeros (Fig. 7c) (which they represent the space between the characters and words) are calculated. The consecutive zeros express two distinct categories: the intra-word distance between the characters and the inter-word distance. The estimation of the threshold that separates these two clusters is achieved minimizing the intra-class variance between them as in the Otsu approach [27]. An example result is shown in Fig. 5h. The outcome of this stage is a list of textual blocks upon which a descriptor will be computed.

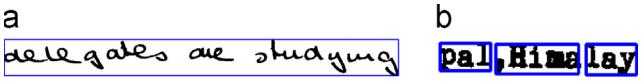

Fig. 6. Example of textual blocks containing: (a) words and (b) parts of a word.

It should be noted that it is not strictly necessary for the success of the proposed method to correctly identify words in each text line. The only requirement, as mentioned earlier, is that the final blocks must be large enough to contain the necessary information for good discrimination performance (Fig. 6).

2.3. Block descriptor extraction

This step involves the creation of the block descriptor using the Bag of Visual Words (BoVW) model.

2.3.1. Codebook creation stage

First of all, a single codebook will accommodate all possible "visual words" that correspond to either the machine printed or the

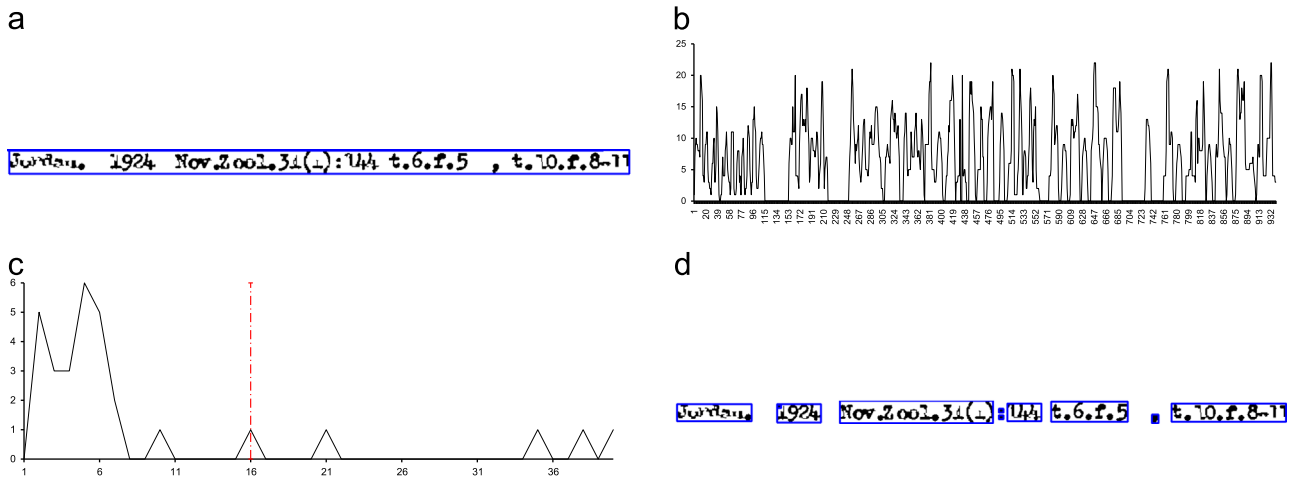

Fig. 7. (a) The initial text line, (b) vertical projection of the text line, (c) histogram of consecutive zeroes along with the calculated threshold, and (d) final text line segmentation.

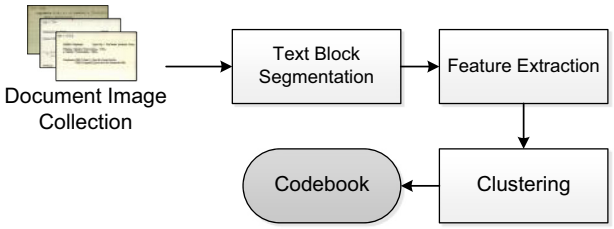

Fig. 8. The steps for codebook creation.

handwritten textual blocks. Fig. 8 shows the individual steps required for the creation of the codebook. After block detection and feature extraction for each block, a clustering is performed. The number of clusters defines the size of the codebook. Predicting the desirable clusters and subsequently the optimal codebook size is non-straightforward and it is dataset-dependent. Generally, it must accommodate the following properties: (i) it must be small enough to ensure a low computational cost through low dimensionality and redundant visual words minimization; (ii) it must be large enough to provide sufficiently high discrimination performance.

In some datasets, the process of trying different sizes of codebooks in order to detect the optimal is feasible, but this is not always the case. The proposed method employs a Self-Growing and Self-Organized Neural Gas (SGONG) network [28] for the detection of the optimal codebook.

The SGONG is a neural classifier that combines the Growing Neural Gas (GNG) [29] network and the cooling learning scheme of the Kohonen Self Organizing Map (KSOM) [30]. It accommodates two separate layers of fully connected neurons, the input layer and the output mapping layer. Contrary to the KSOFM, the space of the mapping layer has always the same dimensionality as the input space. Different from the GNG network, where a new neuron is always inserted at the end of each epoch, the SGONG introduces three new properties that shape adaptively the output lattice of neurons. The properties are: (i) the elimination of the inactive neurons; (ii) the addition of a new neuron, near the one with the maximum contribution in quantization error; (iii) the elimination of the neuron that is close enough to its neighboring neurons

The main benefit of SGONG is the dynamic nature of the output neurons, thus eliminating the need to define the size of codebook in advance. At the end of the process, the output neurons correspond to the set of visual words. In Section 3.3, we evaluate the effectiveness of the proposed codebook creation procedure against different pre-defined sizes calculated by the K-Means algorithm.

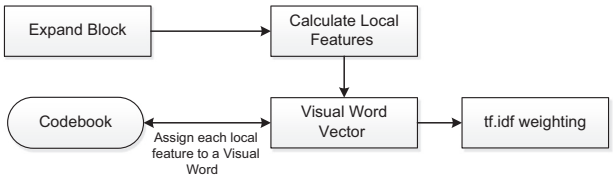

Fig. 9. The creation of the block descriptor.

2.3.2. Block descriptor extraction

After the codebook creation, the calculation of each block descriptor follows. Fig. 9 illustrates the required steps. The local features are calculated on the greyscale version (Fig. 10b) of the original document image. Finally, those keypoints whose corresponding position in the binary image does not match the foreground pixel are rejected (Fig. 10c).

Each of the remaining local features is assigned a Visual Word from the Codebook based on the minimum distance from the center of the corresponding cluster. Finally, a Visual Word Vector is formed based on the appearance of each Visual Word of the Codebook in this particular block. For instance, consider a Codebook with 5 visual words and a block that contains 10 local features which are assigned as follows: 2 local features for the first visual word, 3 local features for the second, 4 local features for the third and 1 local feature for the fifth visual word. Then, the vector which represents the Bag of Visual Words is (2, 3, 4, 0, 1). Note that the dimension of the vector is equal to the number of visual words in the Codebook.

2.3.3. Term frequency and inverse document frequency (tf.idf) weighting

In the classic Bag of Words paradigm in information retrieval theory, a weight is assigned to every word in the dictionary, calculated from the dataset statistics for each codebook word. The most well known and employed statistics are the term frequency (tf) and the inverse document frequency (idf). Previous works which employ tf.idf weighting can be found in image classification [31,32] and retrieval [33,34].

The tf designates the number of appearances of each codebook word in the document while the document frequency df of each word corresponds to the number of documents in the dataset that contains this word. The inverse document frequency of the word w is defined as $idf_w = \log(N/df_w)$, where N is the total number of all the documents and df_w is the document frequency of the word w.

Fig. 10. (a) An example of a text block, (b) initial SIFT keypoints, and (c) final SIFT keypoints.

Table 1
SMART notation for the tf.idf weighting.

tf		df		Normalization	
n (natural)	tf	n (no)	1	n (none)	1
l (logarithm)	$1+\log(tf)$	t (idf)	$\log(N/df)$	c (cosine)	$1/\sqrt{w_1^2+w_2^2+\cdots+w_m^2}$
a (augmented)	$0.5+\frac{0.5xtf}{\max(tf)}$				

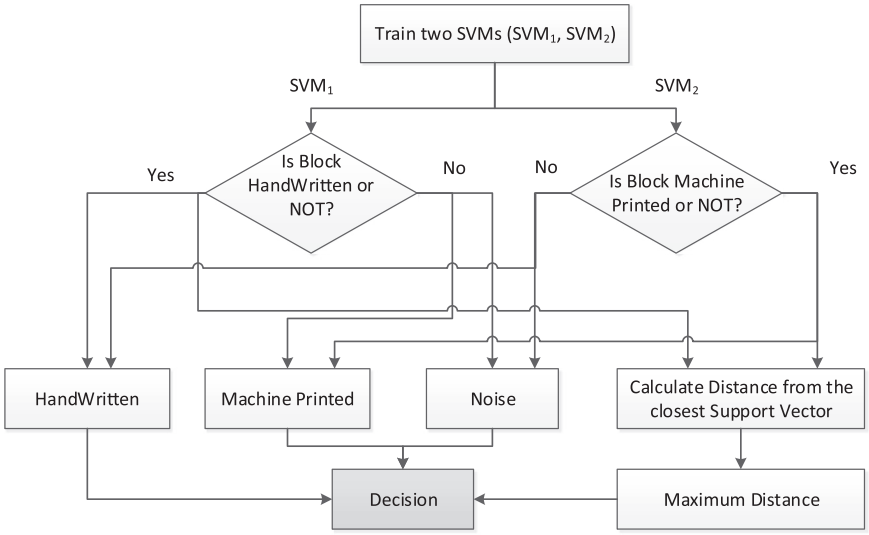

Fig. 11. The classification system algorithm.

Finally, a tf.idf weighting scheme is about combining the tf and idf statistics of a collection. There are many tf.idf weighting variations, so a mnemonic is created for representing a specific combination of weights, which is called SMART notation [35], inspired by the authors of an early text retrieval system. SMART notation represents a combination of weights that takes the form *aaa.bbb* where the first triplet defines the weighting of the document vector and the second designates the weighting of the query. The first letter in each triplet specifies the term frequency component of the weighting, the second the document frequency, and the third the normalization that occurs. As our proposed system is based on a decision algorithm, the query and document weights are the same. Table 1 presents the tf.idf weights variations and their smart notation that have been used in this work in order to investigate their effects to the performance of the proposed system.

In the BoVW, the term frequency corresponds to the visual word frequency in each block while the document frequency corresponds to the number of the blocks wherein each visual word appears. The tf.idf weighting is multiplied to each corresponding element of the vector in order to produce the final block descriptor. Finally, it is worth to note that the tf.idf weighting is a normalized factor, too. In Section 3.4, the evaluation of the effectiveness of each tf.idf weighting is given in order to determine the most suitable for the separation of machine printed and handwritten text.

2.4. Decision system

In this final stage, a classifier decides if the visual word vector of the block contains handwritten or machine printed text or neither of the above (noise). The proposed approach is based on the Support Vector Machines (SVMs) [36,37]. The SVMs are based on statistical learning theory and have been applied to a large number of different classification problems. They are chosen based on their power and their ability that do not require large training sets.

Let D be a given training dataset $[(x_i,y_i)]_{i=1}^{n}$, $x\in[\ 0,1]$, $y\in\{-1,+1\}$, $i\in[1,n]$, where x_i is the ith input vector and y is the label correspond to the x_i. The original linear SVM classifier satisfies the following conditions:

$$\left.\begin{array}{ll} w^Tx_i+b\geq+1 & \text{when } y_i=+1 \\ w^Tx_i+b\leq-1 & \text{when } y_i=-1 \end{array}\right\} \Rightarrow y_i[w^Tx_i+b]-1\geq 0 \quad (1)$$

If the training data are not linearly separable (as in our case) then they mapped from the input space X to a feature space F using kernel. The kernel is transforming the input space to a high dimensional feature space where the training data become linearly separable. For the proposed method the Radial Basis Function (Gaussian) kernel $\exp\{-\gamma\|x-x'\|\}$ is applied. Moreover, the classifier in practice must misclassify some data points (for instance to overcome the over-fitting problem). This is achieved using the

slack variables $\xi_i > 0$. Finally, if $w = \sum_{i=1}^{n} \alpha_i x_i$ Eq. (1) is transformed to: $y_i[\alpha_i k(x, x_i) + b] - 1 + \xi \geq 0$.

Finally, the maximum margin classifier is calculated by solving the following constrained optimization problem which is expressed in terms of variables α_i

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j$$

$$\text{subject to} \quad \sum_{i=1}^{n} y_j \alpha_i = 0, \quad 0 \leq \alpha_i \leq C$$

The constant $C > 0$ defines the trade off between the training error and the margin.

The blocks resulting from the "Text Block Segmentation" stage may contain three types of content: handwritten text, machine-printed text or noise. Therefore, the SVM must classify the block based on the Bag of Visual Words Vector in these three classes.

To achieve this, two binary SVMs are trained as follows: the first (SVM_1) deals with the classification of handwritten text against all the others and the second (SVM_2) deals with the classification of machine printed text against all the others. Fig. 11 illustrates the Classification Scheme. There are four outcomes from the aforementioned SVMs.

- If the SVM_1 output is TRUE and SVM_2 is FALSE then the block contains handwritten text.

Fig. 12. Output examples of the proposed method: Green blocks define machine printed text, blue blocks define handwritten text and red blocks define noise artifacts (a) PRImA-NHM1 dataset, (b) PRImA-UIBK1 dataset, and (c) RRImA-IAM1 dataset. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

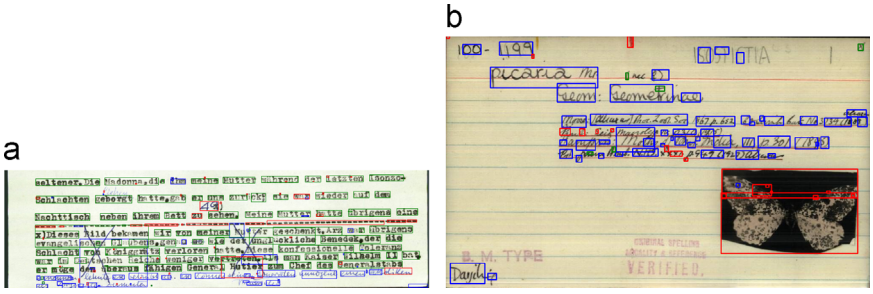

Fig. 13. Failed output examples.

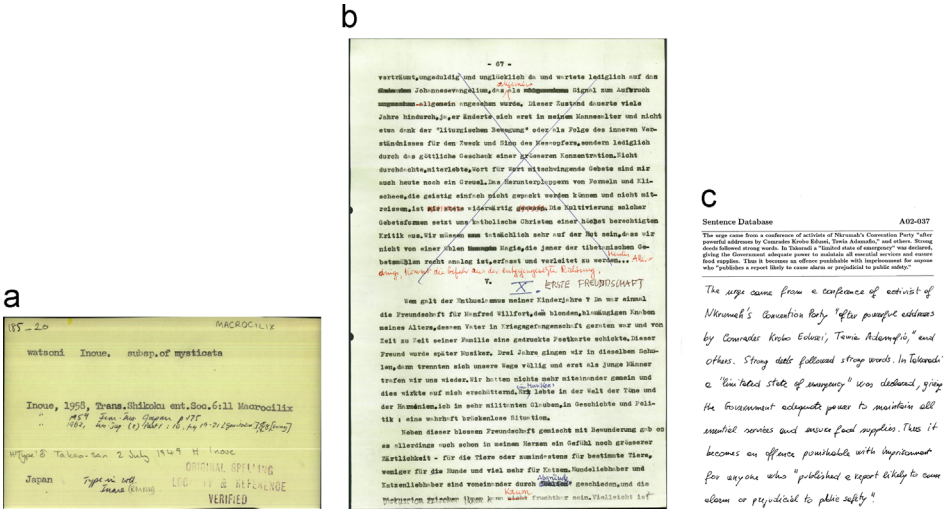

Fig. 14. Representative document examples for the (a) PRImA-NHM1, (b) PRImA-UIBK1, and (c) PRImA-IAM1.

- If the SVM_1 output is FALSE and SVM_2 is TRUE then the block contains machine printed text.
- If the SVM_1 and the SVM_2 output is FALSE then the block contains noise.
- If the SVM_1 and the SVM_2 output is TRUE then the distance of the block descriptor with the closest Support Vector for each SVM_i is calculated. Finally, the class of the block is defined by the SVM_i that is related to the maximum distance among the two aforementioned distances.

The above approach was chosen because the third class which corresponds to noise does not appear frequently compared to the other classes. In many datasets, the blocks that correspond to noise are very few and in many cases there are not enough even for training. In the standard support vector machines for classification, training sets with uneven class sizes result in classification biases towards the class with the large training size [38,36]. Therefore, if a noisy class is defined and the common approaches are used (one-against-all, one-against-one) it may bias the results due to the sheer imbalance between the training samples.

Another advantage of the proposed approach is the training of only two SVMs instead of three SVMs. This reduces the computational cost and considerably increases the speed of the process. Fig. 12 shows the output of the proposed method for a set of document images. Moreover, Fig. 13 shows some failures of the proposed method.

3. Performance

It must be noted that approaches in the literature are using datasets that are not public available. This creates difficulties in performing comparative performance evaluation. Towards solving the aforementioned issue, we provide three distinct datasets that are public available each one containing different machine/handwritten separation context. This removes the existing obstacles in the literature and makes the evaluations of future approaches more easy. The three datasets are

- *100 representative images* selected of the index cards from the *UK Natural History Museum* which contain the scientific names of world Lepidoptera [39]. These cards contain both typewritten and handwritten text. Ground truth was created by the authors. This dataset is denoted as PRImA-NHM1.
- *33 representative typewritten document images* with handwriting annotations. Ground truth was created by the authors. This dataset is denoted as PRImA-UIBK1.
- *103 modified document images* from the *IAM Handwriting Dataset* [40], which comprises forms that contain both handwritten and machine printed English text. The modification applied on the original dataset concerns the removal of the author's name and signature since the IAM dataset does not contain any ground truth for this particular textual information. It is worth to mention that in this work the ground truth of the IAM has been enriched by adding the location of the machine-printed text. This selection is denoted as PRImA–IAM1.

Fig. 14 shows some representative examples for each one of the above datasets.

The ground truth files adhere to the Page Analysis and Ground-truth Elements (PAGE) format framework [41] which is an XML-based representation framework that records detailed information on various aspects of document images and their content. The ground truth files were created using the Aletheia tool [42], an advanced document layout and text ground-truthing system.

For each dataset, we used 15% of randomly chosen documents as training samples (for the creation of the codebook and for training the SVMs) and testing was realized on the remainder (85%). For the SVMs, we used a Radial Basis Function (RBF) kernel. The datasets with the corresponding ground truth files are available freely (see http://datasets.primaresearch.org).

The evaluation of the complete proposed system is not a trivial aspect. For their experimentation most researchers use simple methods [43,44] such as pixel-based or box-based recall, precision measures. Unfortunately, those evaluation strategies have several drawbacks. Firstly, in pixel-based approaches it is very difficult to detect correctly the textual pixel information in the detected objects. Secondly, in box-based approaches the mapping between

Table 2
The weights of the evaluation profile.

Misclassification	2.00	False detection	0.50
Miss	1.00	Merge	0.00
Partial miss	1.00	Split	0.00

Table 3
The overall performance of the proposed system against a baseline method.

Evaluation metric	Dataset					
	PRImA-NHM		PRImA-UIBK1		PRImA-IAM	
	Character F-measure	PRImA evaluation	Character F-measure	PRImA evaluation	Character F-Measure	PRImA evaluation
Gabor features	0.614	0.706	0.886	0.862	0.880	0.889
Proposed system	**0.844**	**0.842**	**0.928**	**0.922**	**0.989**	**0.989**

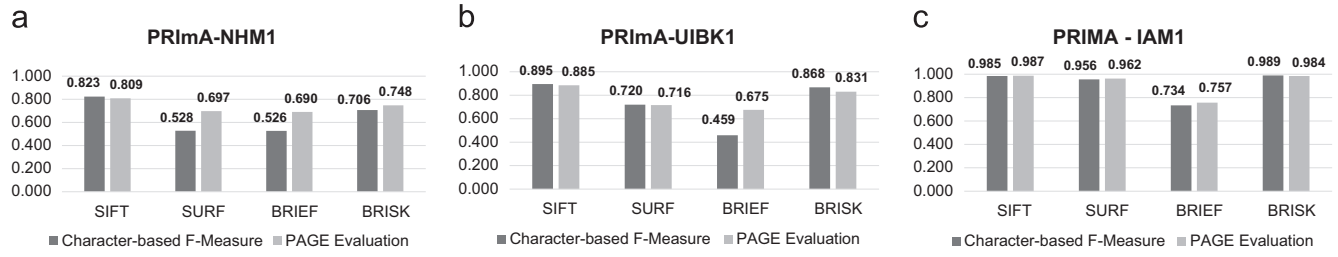

Fig. 15. Effectiveness of the proposed method based on different local features: (a) PRImA-NHM1 dataset, (b) PRImA-UIBK1 dataset, and (c) PRImA-IAM1 dataset.

ground truth and detected objects can mislead results in the case of splits and/or mergers.

To overcome these problems, we employ two different evaluation methods. We consider this to be of considerable benefit to the readers of this paper to present, use and compare two complementary evaluation methodologies that are state-of-the-art. The first one is the PRImA Layout Evaluation Framework [45] which has been previously used at the ICDAR2011 Historical Document Layout Analysis Competition [46]. In this framework, all the bounding boxes (ground truth and method results) for a given document image are transformed into an interval representation, which allows efficient comparison and calculation of overlapping/missed parts. The common occurrences between the produced representations are determined and finally errors are identified, quantified and qualified based on an evaluation profile [45]. For the purpose of the proposed method evaluation, a profile with weights for Misclassification, Miss/Partial Miss and also False Detection is used. Table 2 shows the weights used for the construction of the evaluation profile. The most important penalty in this case is misclassification and therefore it has the largest weight. The additional weights are set to evaluate the overall performance of the method, not just the performance of classification. Missing (not detecting at all) or partially missing a component is given a normal weight while detecting a component that does not exist is given a lower penalty as this is not so relevant in this application scenario. All the weights can be reconfigured but those values produce a good indication of failure/success from experience. Worth to note that this method is based on the comparison of bounding polygons and their black pixels.

The second evaluation method used is the estimated character-based F-measure [47] technique. This technique is complementary to the one above as it is not based on geometric properties (outline) of a block and its black pixels but attempts to base its evaluation on the number of detected characters within each block. Although, the detection of the character number is a hard task to achieve, it can be approximately identified by the ratio width/height of the box if the assumption holds that this ratio does not vary for every character, the spaces between different words in a block are proportional to its height and each block contains characters of the same size.

The overall metric is a weighted harmonic mean of precision and recall (Eq. (2)).

$$F_{ecn} = \frac{2 * Precision_{ecn} * Recall_{ecn}}{Precision_{ecn} + Recall_{ecn}} \tag{2}$$

Table 4
The number of clusters K used for the K-means clustering.

Dataset	Codebook 2	Codebook 3	Codebook 4
PRImA-NHM1	247	150	500
PRImA-UIBK1	626	500	1000
PRImA-IAM1	449	200	500

$$Recall_{ecn} = \frac{\sum_{i=1}^{N} \frac{|GDI_i|}{hg_i^2}}{\sum_{i=1}^{N} \frac{|GB_i|}{hg_i^2}}, \quad Precision_{ecn} = \frac{\sum_{i=1}^{N} \frac{|DGI_i|}{hg_i^2}}{\sum_{i=1}^{N} \frac{|DB_i|}{hg_i^2}} \tag{3}$$

where GB_i denotes the ground truth bounding box number i with height hg_i, DB_i denotes the detected bounding box number i with height hd_i, N denotes the number of ground truth bounding boxes, M denotes the number of detected bounding boxes and GDI, DGI are calculated as in the following equations:

$$GDI_i = GB_i \bigcap \left(\bigcup_{i=1}^{M} DB_i \right), \quad DGI_i = DB_i \bigcap \left(\bigcup_{i=1}^{M} GB_i \right) \tag{4}$$

The GB_i and the DB_i are the skeletonization images of the ground truth and detected boxes i, respectively. It is worth to note that GDI_i and DGI_i are numbers defined by the common foreground pixels between the detected and ground truth bounding boxes. More detailed mathematical explanation how the following equations are calculated is in [47]. The above performance evaluation metric is based on the intersection of the ground truth and the resulting bounding boxes, normalized by the estimated number of contained characters therefore, it estimates more objectively each text separation system.

3.1. Overall performance

Table 3 shows the overall performance of the proposed method for the three aforementioned datasets. The selected local features are the SIFT (Section 3.2), the codebook is created using the SGONG classifier (Section 3.3), the n.n.c tfidf weight is applied (Section 3.4) and finally, the output is determined from the proposed classification system algorithm (Section 3.5). Moreover, to showcase the advantage of the BoVW approach we evaluate it against a baseline system using the Gabor Features [48] as implemented in [49]. That means that the segmentation and decision stage is the same, just the BoVW system with the corresponding local features and tfidf weighting is replaced with the common Gabor Features. Table 3 shows that the proposed system has superior performance against a baseline procedure. Furthermore, the output of the two different evaluation metrics are shown to be similar and therefore in future evaluations one of those two measures is adequate for consistent performance evaluation.

The next experiments evaluate each distinct component.

3.2. Evaluating local features

This section describes the evaluation of different local features and their impact in the effectiveness for the proposed model. The examined local features are: SIFT [14], SURF [15], BRIEF [16], BRISK [17] which as described in Section 2.1 they have different properties and strengths. Based on the graphs showing in Fig. 15, SIFT outperforms the local features so it is the most appropriate local feature for the proposed BoVW model. The binary BRISK feature

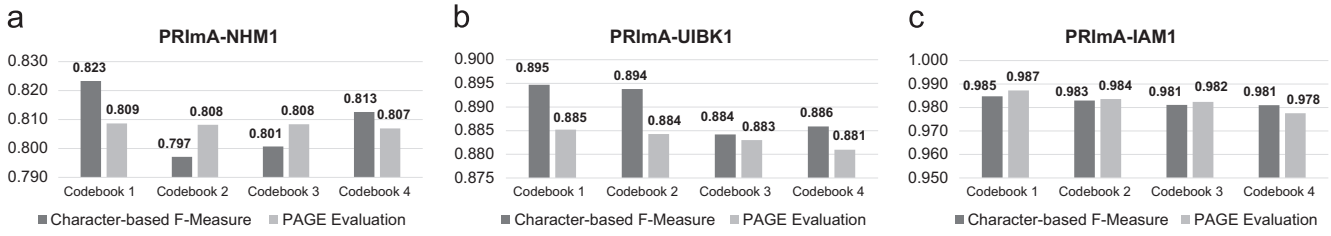

Fig. 16. Effectiveness of the proposed method based on codebook: (a) PRImA-NHM1 dataset, (b) PRImA-UIBK1 dataset, and (c) PRImA-IAM1 dataset.

ranked second in all datasets making them a good choice if the method speed costs must be considered.

3.3. Evaluating the Codebook

As discussed earlier in Section 2.3.1, the codebook size is a vital component in the BoVW model. This section investigates the correlation between the visual words number and the effectiveness of the proposed method and clarifies the advantages of the SGONG neural net.

For each dataset, four codebooks have been created which are based on: (i) the clusters automatically determined by the SGONG neural net (Codebook 1); (ii) the clusters determined by the K-Means where K equals to "Codebook 1" size (Codebook 2); (iii) the clusters determined by K-Means where K has been randomly chosen by a value below the K of "Codebook 2" (Codebook 3); (iv) the clusters determined by K-Means where K has been randomly chosen by a value above the K of "Codebook 2" (Codebook 4). The number of clusters K used for the K-Means clustering is shown in Table 4.

The results show in (Fig. 16a, b and c) indicate that the SGONG neural net provides a two-fold advantage. On one hand it successfully detects the optimal number of the visual words and on the other hand, the classes produced are better from those by K-Means.

Table 5
The effect of the tf.idf weighting for the PRImA-NHM1 dataset.

tf.idf weighting	SIFT F-Measure/PRImA	Increase/decrease F-Measure/PRImA	BRISK F-Measure/PRImA	Increase/decrease F-Measure/PRImA
n.n.n	0.823/0.809	–	0.706/0.748	–
n.n.c.	0.839/0.841	1.92%/3.99%	**0.730/0.781**	**3.31%/4.45%**
n.t.n	0.806/0.804	−2.13%/−0.59%	0.702/0.745	−0.58%/−0.38%
n.t.c	**0.844/0.842**	**2.56%/4.16%**	0.712/0.764	0.74%/2.17%
l.n.n	0.787/0.803	−4.43%/−0.72%	0.688/0.761	−2.66%/1.67%
l.n.c	0.842/0.837	2.30%/3.54%	0.711/0.762	0.59%/1.88%
l.t.n	0.811/0.808	−1.53%/−0.10%	0.706/0.747	−0.11%/−0.13%
l.t.c	**0.843/0.836**	**2.42%/3.39%**	0.702/0.759	−0.58%/1.47%
a.n.n	0.827/0.829	0.41%/2.56%	0.698/0.752	−1.25%/0.57%
a.n.c	0.838/0.830	1.74%/2.66%	0.742/0.761	5.00%/1.72%
a.t.n	0.833/0.819	1.19%/1.25%	0.713/0.744	0.91%/−0.53%
a.t.c	0.835/0.822	1.40%/1.64%	0.745/0.763	5.41%/1.92%

Table 6
The effect of the tf.idf weighting for the PRImA-UIBK1 dataset.

tf.idf weighting	SIFT F-Measure/PRImA	Increase/decrease F-Measure/PRImA	BRISK F-Measure/PRImA	Increase/decrease F-Measure/PRImA
n.n.n	0.895/0.885	–	0.868/0.831	–
n.n.c.	**0.928/0.922**	**3.72%/4.12%**	**0.877/0.856**	**1.03%/3.07%**
n.t.n	0.892/0.866	−0.31%/−2.19%	0.871/0.831	0.35%/0.10%
n.t.c	0.930/0.933	3.93%/5.43%	0.849/0.819	−0.17%/−1.43%
l.n.n	0.909/0.892	1.60%/0.73%	0.870/0.833	0.21%/0.28%
l.n.c	0.928/0.922	3.70%/4.12	0.875/0.853	0.80%/2.64%
l.t.n	0.890/0.866	−0.53%/−2.20%	0.866/0.833	−0.24%/0.32%
l.t.c	**0.927/0.918**	**3.61%/3.74%**	0.860/0.843	−0.89%/1.46%
a.n.n	0.902/0.882	0.82%/−0.36%	0.874/0.845	0.76%/1.69%
a.n.c	0.866/0.839	−3.22%/−5.22%	0.795/0.765	−8.34%/−7.89%
a.t.n	0.901/0.879	0.67%/−0.72%	0.864/0.834	−0.45%/0.47%
a.t.c	0.886/0.884	−0.96%/−0.12%	0.827/0.797	−4.74%/−4.04%

Table 7
The effect of the tf.idf weighting for the PRImA-IAM1 dataset.

tf.idf weighting	SIFT F-Measure/PRImA	Increase/decrease F-Measure/PRImA	BRISK F-Measure/PRImA	Increase/decrease F-Measure/PRImA
n.n.n	0.985/0.987	–	0.989 0.984	–
n.n.c.	**0.989/0.989**	**0.47% 0.16%**	**0.992 0.989**	**0.22%/0.42%**
n.t.n	0.987/0.988	0.25%/0.05%	0.990/0.985	0.03%/0.09%
n.t.c	0.989/0.989	0.47%/0.18%	0.989/0.981	−0.06%/−0.34
l.n.n	0.988/0.986	0.30%/−0.18%	0.989/0.984	−0.04%/−0.03
l.n.c	0.990/0.990	0.54%/0.31%	0.992/0.989	0.23%/0.42%
l.t.n	0.987/0.985	0.22%/−0.20	0.990/0.985	0.02%/0.03%
l.t.c	**0.990/0.991**	**0.53%/0.34%**	0.992/0.988	0.21%/0.38%
a.n.n	0.988/0.988	0.37%/0.07%	0.990/0.984	0.02%/−0.06%
a.n.c	0.988/0.987	0.29%/−0.05%	0.989/0.981	−0.04%/−0.33%
a.t.n	0.989/0.989	0.47%/0.19%	0.990/0.983	0.01%/−0.14%
a.t.c	0.989/0.988	0.39%/0.09	0.990/0.984	0.05.%/−0.04%

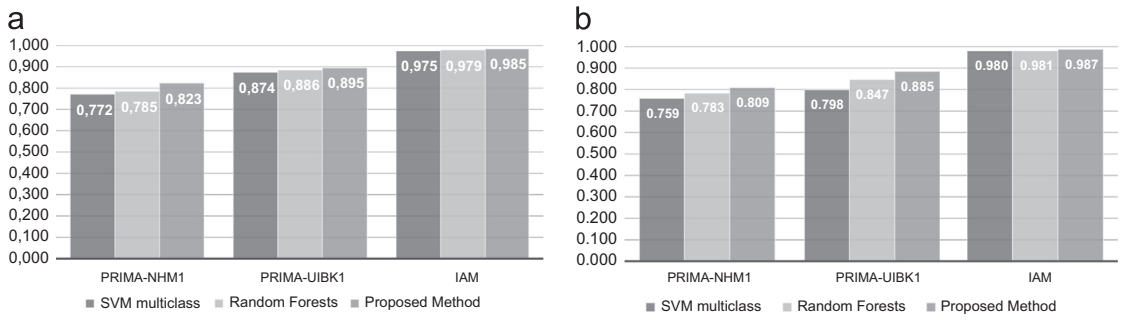

Fig. 17. The performance evaluation of different decision systems (a) with Character-based F-Measure evaluation metric, (b) PRImA Evaluation Framework.

3.4. Evaluating tf.idf weighting

The next experiment involves the exploration of the inducing effect of different tf.idf weighting schemes in the performance of the proposed BoVW model. Moreover, in order to investigate the tf.idf weight under different scenarios, both the SIFT and BRISK local features are employed to three different datasets. Table 5 shows the results for the PRImA-NHM1, Table 6 for the PRImA-UIBK1 and Table 7 for the PRImA-IAM1 dataset. For the SIFT local features the l.t.c. and n.n.c tfidf weighting causes some robust increase in performance across the three datasets. While for the BRISK local features the n.n.c tfidf weight is the only robust choice for the three datasets.

3.5. Evaluating against different decision systems

Concerning the investigation of the proposed decision system performance, we evaluate it against a multi-class Support Vector Machine [50] and the Random forests [51] machine learning algorithms. The SVM multiclass [50] does not break down the problem into multiple independent binary classification tasks. Instead, it tries to transform the problem into multiple optimization reduced size goals. The Random forests [51] are a synthesis of multiple tree predictors, in which each tree is created from a random subset of the train data that have the same distribution with it.

Fig. 17 depicts the results of the evaluation experiments. They show that the proposed method performs better, the Random forests rank second and third the SVM multiclass.

4. Conclusion

In this paper, a method based on the Bag of Visual Words paradigm for the separation of the machine printed and handwritten text is presented. It is a generic approach which can deal with document images which originate from datasets that are situated into different machine-printed/handwritten separation context. The proposed BoVW model is coupled with an optimal codebook creation using a Self-Growing and Self-Organized Neural Gas (SGONG) network. For this model, it is shown that among several state of the art local features, SIFT achieves the best performance. The performance evaluation relies upon two distinct methodologies. The first methodology is based on the number of estimated characters within each block while the other is based on geometric properties. It is worth to note that both evaluation methods resulted in a mutual agreement at each comparative experiment. Last but not least, we provide as public available three distinct datasets (each one containing different machine printed/handwritten separation context) including the corresponding ground truth.

Conflict of interest statement

None declared.

References

[1] V. Govindan, A. Shivaprasad, Character recognition – a review, Pattern Recognition 23 (7) (1990) 671–683.
[2] J.K. Guo, M.Y. Ma, Separating handwritten material from machine printed text using hidden Markov models, in: International Conference on Document Analysis and Recognition, 2001, p. 0439.
[3] U. Pal, B.B. Chaudhuri, Machine-printed and hand-written text lines identification, Pattern Recognition Letters 22 (3-4) (2001) 431–441.
[4] U. Pal, B. Chaudhuri, Automatic separation of machine-printed and hand-written text lines, in: Proceedings of the Fifth International Conference on Document Analysis and Recognition, ICDAR '99, 1999, pp. 645–648.
[5] E. Kavallieratou, S. Stamatatos, Discrimination of machine-printed from hand-written text using simple structural characteristics, Pattern Recognition 1 (2004) 437–440.
[6] J. Eduardo Bastos Dos Santos, B. Dubuisson, F. Bortolozzi, Characterizing and distinguishing text in bank cheque images, in: Proceedings XV Brazilian Symposium on Computer Graphics and Image Processing, IEEE, 2002, pp. 203–209.
[7] L. da Silva, A. Conci, A. Sanchez, Automatic discrimination between printed and handwritten text in documents, in: 2009 XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI), 2009, pp. 261–267.
[8] F. Farooq, K. Sridharan, V. Govindaraju, Identifying handwritten text in mixed documents, in: Proceedings of 18th International Conference on Pattern Recognition (ICPR) 2006, vol. 2, 2006, pp. 1142–1145.
[9] X. Peng, S. Setlur, V. Govindaraju, R. Sitaram, Handwritten text separation from annotated machine printed documents using Markov random fields, International Journal on Document Analysis and Recognition (2011) 1–16.
[10] Y. Zheng, H. Li, D. Doermann, Machine printed text and handwriting identification in noisy document images, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (3) (2004) 337–353.
[11] K.C. Fan, L.S. Wang, Y.T. Tu, Classification of machine-printed and handwritten texts using character block layout variance, Pattern Recognition 31 (9) (1998) 1275–1284.
[12] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, 2004, p. 22.
[13] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Foundations and Trends in Computer Graphics and Vision 3 (2008) 177–280.
[14] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.
[15] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Computer Vision and Image Understanding 110 (3) (2008) 346–359.
[16] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, P. Fua, Brief: computing a local binary descriptor very fast, IEEE Transactions on Pattern Analysis and Machine Intelligence, (99) (2011) 1–1.
[17] S. Leutenegger, M. Chli, R. Siegwart, Brisk: binary robust invariant scalable keypoints, in: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2548–2555.
[18] A. Bolovinou, I. Pratikakis, S. Perantonis, Bag of spatio-visual words for context inference in scene classification, Pattern Recognition 46 (3) (2013) 1039–1053.
[19] M. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 1447–1454.
[20] T. Deselaers, L. Pimenidis, H. Ney, Bag-of-visual-words models for adult image classification and filtering, in: ICPR, 2008, pp. 1–4.
[21] L. Rothacker, S. Vajda, G.A. Fink, Bag-of-features representations for offline handwriting recognition applied to arabic script, in: 2012 International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE, 2012, pp. 149–154.

[22] R. Shekhar, C. Jawahar, Word image retrieval using bag of visual words, in: 2012 10th IAPR International Workshop on Document Analysis Systems (DAS), 2012, pp. 297–301.
[23] B. Gatos, I. Pratikakis, S. Perantonis, Adaptive degraded document image binarization, Pattern Recognition 39 (3) (2006) 317–327.
[24] P.V. Hough, Method and Means for Recognizing Complex Patterns, Us Patent 3,069,654, December 18, 1962.
[25] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, Incorporated, 2008.
[26] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, N. Papamarkos, Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths, Image and Vision Computing 28 (4) (2010) 590–604.
[27] N. Otsu, A threshold selection method from gray-level histograms, Automatica 11 (285–296) (1975) 23–27.
[28] A. Atsalakis, N. Papamarkos, Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural gas, Engineering Applications of Artificial Intelligence 19 (7) (2006) 769–786.
[29] B. Fritzke, et al., A growing neural gas network learns topologies, Advances in neural information processing systems 7 (1995) 625–632.
[30] T. Kohonen, The self-organizing map, Proceedings of the IEEE 78 (9) (1990) 1464–1480.
[31] J. Yang, Y. Jiang, A. Hauptmann, C. Ngo, Evaluating bag-of-visual-words representations in scene classification, in: Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval, ACM, 2007, pp. 197–206.
[32] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 2006, pp. 2169–2178.
[33] S. Chatzichristofis, C. Iakovidou, Y. Boutalis, O. Marques, Co.vi.wo.: color visual words based on non-predefined size codebooks, IEEE Transactions on Cybernetics 43 (1) (2013) 192–205, http://dx.doi.org/10.1109/TSMCB.2012.2203300.
[34] W. Zhao, Y. Jiang, C. Ngo, Keyframe retrieval by keypoints: can point-to-point matching help?, Image and Video Retrieval 4071 (2006) 72–81.
[35] C. Manning, P. Raghavan, H. Schutze, Introduction to Information Retrieval, vol. 1, Cambridge University Press Cambridge, 2008.
[36] C. Cortes, V. Vapnik, Support vector networks, Machine Learning 20 (1995) 197–273.
[37] B.E. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: COLT, 1992, pp. 144–152.
[38] V. Vapnik, An overview of statistical learning theory, IEEE Transactions on Neural Networks 10 (5) (1999) 988–999, http://dx.doi.org/10.1109/72.788640.
[39] G. Beccaloni, M. Scoble, L. Kitching, T. Simonsen, G. Robinson, B. Pitkin, A. Hine, The Global Lepidoptera Names Index (lepindex), World Wide Web Electronic Publication ⟨http://www.nhm.ac.uk/entomology/lepindex⟩(accessed 12 March 2012).
[40] U. Marti, H. Bunke, The iam-database: an english sentence database for offline handwriting recognition, International Journal on Document Analysis and Recognition 5 (1) (2002) 39–46.
[41] S. Pletschacher, A. Antonacopoulos, The page (page analysis and ground-truth elements) format framework, in: 2010 20th International Conference on Pattern Recognition (ICPR), IEEE, 2010, pp. 257–260.
[42] C. Clausner, S. Pletschacher, A. Antonacopoulos, Aletheia-an advanced document layout and text ground-truthing system for production environments, in: 2011 International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2011, pp. 48–52.
[43] M. Lyu, J. Song, M. Cai, A comprehensive method for multilingual video text detection, localization, and extraction, IEEE Transactions on Circuits and Systems for Video Technology 15 (2) (2005) 243–255.
[44] C. Jung, Q. Liu, J. Kim, A stroke filter and its application to text localization, Pattern Recognition Letters 30 (2) (2009) 114–122.
[45] C. Clausner, S. Pletschacher, A. Antonacopoulos, Scenario driven in-depth performance evaluation of document layout analysis methods, in: 2011 International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2011, pp. 1404–1408.
[46] A. Antonacopoulos, C. Clausner, C. Papadopoulos, S. Pletschacher, Historical document layout analysis competition, in: 2011 International Conference on Document Analysis and Recognition (ICDAR), IEEE, 2011, pp. 1516–1520.
[47] M. Anthimopoulos, B. Gatos, I. Pratikakis, A two-stage scheme for text detection in video images, Image and Vision Computing 28 (9) (2010) 1413–1426.
[48] I. Fogel, D. Sagi, Gabor filters as texture discriminator, Biological Cybernetics 61 (2) (1989) 103–113, http://dx.doi.org/10.1007/BF00204594.
[49] K. Zagoris, I. Pratikakis, A. Antonacopoulos, B. Gatos, N. Papamarkos, Handwritten and machine printed text separation in document images using the bag of visual words paradigm, in: 2012 International Conference on Frontiers in Handwriting Recognition (ICFHR), 2012, pp. 103–108. http://dx.doi.org/10.1109/ICFHR.2012.207.
[50] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, The Journal of Machine Learning Research 2 (2002) 265–292.
[51] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

Konstantinos Zagoris received the Diploma in Electrical and Computer Engineering in 2003 from Democritus University of Thrace, Greece and his PhD on Content and Metadata Based Image Document Retrieval from the same university in 2010. He worked as a Research Assistant at Salford University in 2011. He is currently working as a Post-Doctoral Researcher at the Democritus University of Thrace. His research interests include document image retrieval, colour image processing and analysis, document analysis, pattern recognition, databases and operating systems. He is author of more than 20 publications in journals and international conference proceedings and has participated in several research programs funded by the Hellenic Republic or European community. He is a member of the Technical Chamber of Greece.

Ioannis Pratikakis is an Assistant Professor at the Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece. He received the Ph. D. degree in Applied Sciences from the Department of Electronics Engineering and Computer Science at Vrije Universiteit Brussel, Belgium, in January 1999. From March 1999 to March 2000, he was at IRISA, Rennes, France as an INRIA postdoctoral fellow. Since 2003, he was working as Research Scientist at the Institute of Informatics and Telecommunications in the National Centre for Scientific Research "Demokritos", Athens, Greece. His research interests include 2D & 3D image processing and analysis, 2D & 3D image sequence analysis, document image analysis and recognition, 3D computer vision, graphics and multimedia search and retrieval with a particular focus on visual modalities.

Apostolos Antonacopoulos leads the Pattern Recognition and Image Analysis research Lab at the School of Computing, Science and Engineering at the University of Salford, UK where he currently holds the post of Senior Lecturer. He received his PhD from the University of Manchester, Institute of Science and Technology (UMIST), UK in 1995. From 1995 to 2004 he worked as a Lecturer in the Department of Computer Science at the University of Liverpool where he founded PRImA. In 2005, he received the IAPR/ICDAR Young Investigator Award for "Outstanding service to the ICDAR community and his innovative research in historical document processing applications."

Basilis Gatos was born in 1967, in Athens, Greece. His Ph.D. thesis is on Optical Character Recognition Techniques. In 1993 he was awarded a scholarship from the Institute of Informatics and Telecommunications, NCSR "Demokritos", where he worked till 1996. From 1997 to 1998 he worked as a Software Engineer at Computer Logic S.A. From 1998 to 2001 he worked at Lambrakis Press Archives as a Director of the Research Division in the field of digital preservation of old newspapers. From 2001 to 2003 he worked at BSI S.A. as Managing Director of R&D Division in the field of document management and recognition. He is currently working as a Researcher at the Institute of Informatics and Telecommunications of the National Center for Scientific Research "Demokritos", Athens, Greece. His main research interests are in Image Processing and Document Image Analysis, OCR and Pattern Recognition.

Nikos Papamarkos was born in Alexandroupoli, Greece, in 1956. He received his Diploma Degree in Electrical and Mechanical Engineering from the University of Thessaloniki, Thessaloniki, Greece, in 1979 and the Ph.D. Degree in Electrical Engineering in 1986, from the Democritus University of Thrace, Greece. He was elected as a Lecturer (1987–1990) and then promoted to Assistant Professor (1990–1996), Associate Professor (1996–2003) and Professor (2003–now) at the Democritus University of Thrace. During 1987 and 1992 he has also served as a Visiting Research Associate at the Georgia Institute of Technology, USA. His current research interests lie in digital image processing, computer vision, document processing, analysis and recognition, pattern recognition, neural networks, signal processing, filter design and optimization algorithms. He is a Senior Member of IEEE, Member of IAPR, Member of IEEE and Member of the Greek Technical Chamber.