

Article

Text/Non-Text Separation from Handwritten Document Images Using LBP Based Features: An Empirical Study

Sourav Ghosh ^{1,*}, Dibyadwati Lahiri ^{1,*}, Showmik Bhowmik ¹, Ergina Kavallieratou ² and Ram Sarkar ¹

¹ Department of Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal 700032, India; showmik.cse@gmail.com (S.B.); raamsarkar@gmail.com (R.S.)

² Department of Information and Communication Systems Engineering, University of Aegean, Lesbos 811 00, Greece; kavallieratou@aegean.gr

* Correspondence: souravghosh2197@gmail.com (S.G.); dibyadwati.lahiri@gmail.com (D.L.)

Received: 15 December 2017; Accepted: 6 April 2018; Published: 12 April 2018



Abstract: Isolating non-text components from the text components present in handwritten document images is an important but less explored research area. Addressing this issue, in this paper, we have presented an empirical study on the applicability of various Local Binary Pattern (LBP) based texture features for this problem. This paper also proposes a minor modification in one of the variants of the LBP operator to achieve better performance in the text/non-text classification problem. The feature descriptors are then evaluated on a database, made up of images from 104 handwritten laboratory copies and class notes of various engineering and science branches, using five well-known classifiers. Classification results reflect the effectiveness of LBP-based feature descriptors in text/non-text separation.

Keywords: text/non-text separation; local binary pattern; handwritten document; document image processing; texture-based features

1. Introduction

Documents, in the modern day, are required to be stored in digitized form to increase their longevity, portability and security. In order to achieve this purpose, the development of a complete Document Image Processing System (DIPS) has become an utmost need. Along with the other steps, any DIPS needs to identify the texts present in a document image separately from the non-text components like tables, diagrams, graphic designs before processing the text through an Optical Character Recognition (OCR) engine [1–3]. The reason for this is very obvious: OCR engines do not process non-text components. Researchers, to date, have reported many solutions to this problem for printed documents [4–6]. However, the same is not true for regular handwritten documents; a rather limited amount of work is available in this area, to the best of our knowledge, among which two significant ones are [7,8]. In document image processing, researchers mostly use OCR technology in order to work on word and/or character level to provide a viable solution for information content exploitation [9].

In general, handwritten documents are unstructured i.e., in most cases, these documents do not follow any specific layout, unlike the printed documents. Thus, the appearance of text and non-text in handwritten documents is very chaotic. For example, text components often overlap with the non-text components. Furthermore, the building blocks (i.e., characters) of the text in handwritten documents do not follow the standard shape and size usually found in its printed counterpart. One of the key difficulties in the graphics recognition domain is also to work on complex and composite symbol

recognition, retrieval and spotting [10]. Thus, the separation of text and non-text in handwritten documents is comparably complex than in printed documents.

Mostly, the reported solutions to the problem of text and non-text separation are done either at the region level [4] or at the connected component (CC) level [5,6]. Methods that implement text/non-text separation at the region level initially perform region segmentation and then classify each segmented region as either a text or graphics region. For classifying the segmented regions, researchers have mostly used texture based features like Gray Level Co-occurrence Matrix (GLCM) [4,11] Run-length based features [12,13] or white tiles based features [14]. However, region segmentation based methods are very sensitive to the segmentation results. Poor segmentation can cause a significant degradation in the classification result. On the other hand, as CC based methods work at the component level, they do not suffer from such a problem. Methods that follow a CC based approach use shape-based features [5,6]. In general, methods reported in this literature for text/non-text separation in handwritten documents have mostly followed the CC based approach [7,8]. It is worth mentioning here that, as historical handwritten manuscripts suffer from various quality degradation issues, techniques like binarization and CC extraction become very error prone. Thus, in some recent articles [15–18], researchers have followed a pixel based approach, which avoids the binarization and CC extraction steps.

From the available research work on this topic, it can be observed that texture features like GLCM (Gray Level Co-occurrence matrix) [4,11], Run-length encoding based features [12,13], Black-and-white transitional matrix based feature [19] have been commonly used by researchers to solve the text/non-text separation problem for printed documents, as well as to separate handwritten and printed text sections in documents [20]. In a recent work [8], a Rotation Invariant Uniform Local Binary Pattern (RIULBP) operator has also been used successfully to separate the text and non-text components in handwritten class-notes. Texture features have proven to be very useful in the field of text/non-text separation due to the fact that text regions and graphics regions in most cases have very different patterns, which can be exploited to differentiate between them. Motivated by this fact, in the present work, we have attempted to evaluate the performance of different Local Binary Pattern (LBP) based texture features to classify the components present in handwritten documents as text or non-text.

The key contributions of our paper are as follows:

1. We have given a detailed analysis of how accurately features extracted by different variants of the LBP operator from handwritten document images help in differentiating text components from non-text ones, which is one of the most challenging research areas in the domain of document image processing. For that purpose, we have considered five variants of LBP [21], namely, the basic LBP [22], improved LBP [23], rotation invariant LBP [22], uniform LBP [22], and rotation invariant and uniform LBP [22].
2. The contents of the dataset, used here for evaluation, have complex text and non-text components as well as variations in terms of scripts, as we have considered both Bangla and English texts. In addition to that, some of the documents have handwritten as well as printed texts.
3. We have also made a minor alteration to robust LBP [24] in order to develop robust and uniform LBP. A method to determine the appropriate threshold value used in this variant of LBP for handwritten documents has also been proposed.

2. Local Binary Patterns and Its Variants

LBP was first introduced by Ojala [25,26], as a computationally simple texture operator in a monochrome texture image.

The generalized definition of LBP, given in [22], used M sample points evenly placed on a circle of radius R with its center positioned at (x_{cen}, y_{cen}) . The position (x_p, y_p) of the neighboring point p , where $p \in 0, 1, \dots, M - 1$ is given by

$$(x_p, y_p) = (x_{cen} + R \cos(2\pi p/M), y_{cen} - R \sin(2\pi p/M)). \quad (1)$$

Let T be the feature vector representing the local texture:

$$T = \text{func}(I_{cen}, I_0, I_1, \dots, I_{M-1}),$$

where I_{cen} and I_p for $p \in \{0, 1, \dots, M-1\}$ represent gray values of the center pixel and the neighboring pixels, respectively. To achieve gray scale invariance, the texture operator is modified to consider the *difference* in intensities of the center pixel and its neighbors:

$$T = \text{func}(I_0 - I_{cen}, I_1 - I_{cen}, \dots, I_{M-1} - I_{cen}).$$

Furthermore, to achieve a robustness against the scaling of grayscale, only the signs of difference in intensities are considered:

$$T = \text{func}(f(I_0 - I_{cen}), f(I_1 - I_{cen}), \dots, f(I_{M-1} - I_{cen})).$$

Here,

$$f(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0. \end{cases} \quad (2)$$

Finally, the LBP operator, for the center pixel p^{cen} having intensity value I_{cen} with M neighbors (X_1, X_2, \dots, X_M) of intensities (I_1, I_2, \dots, I_M) , respectively, can be defined below:

$$LBP_{(M,R)}(x_{cen}, y_{cen}) = \sum_{n=1}^M f(I_n - I_{cen}) \times 2^{n-1}. \quad (3)$$

LBP creates an M -bit string. Hence, for $M = 8$, the values of $LBP_{(M,R)}(x^{cen}, y^{cen})$ can vary from 0 to 255. The process is depicted in Figure 1.

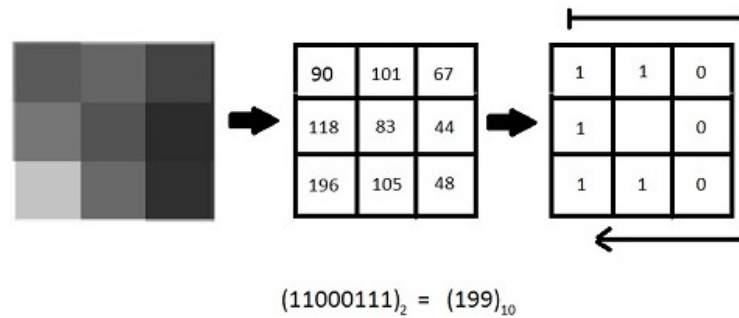


Figure 1. Illustration of LBP value generation for a 3×3 gray image window, where $M = 8$, and radius = 1.

In order to efficiently extract texture features of various complexities, the original LBP operator has been modified to generate a number of variants.

2.1. Improved LBP (ILBP)

The main difference between ILBP [23] and simple LBP is that, instead of the intensity of the center pixel, the mean intensity value of all the pixels, including the center pixel, is used to find the intensity difference during binary pattern computation. In addition to that, while computing ILBP, the intensity of the center pixel is also compared with mean intensity. ILBP is formally defined as follows:

$$ILBP_{(M,R)}(x_{cen}, y_{cen}) = \sum_{n=0}^{M-1} f(I_n - I_{mean}) \times 2^n + f(I_{cen} - I_{mean}) \times 2^M, \quad (4)$$

$$I_{mean} = \frac{(\sum_{n=0}^{M-1} I_n) + I_{cen}}{M + 1}. \quad (5)$$

The value of $f(x)$ is computed as given in Equation (2). As ILBP additionally considers the center pixel, thus the value of $ILBP_{(M,R)}(x_{cen}, y_{cen})$ can vary from 1 to 511 (see Figure 2).

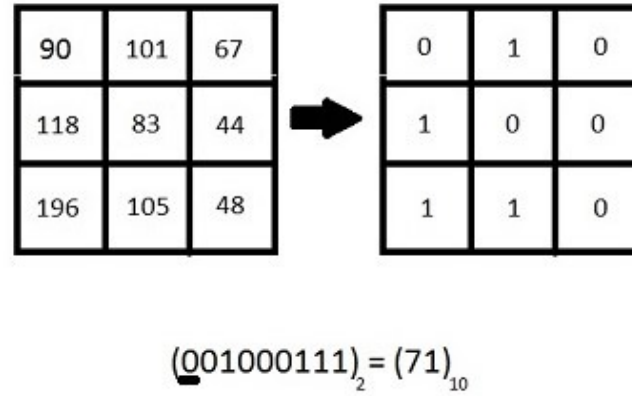


Figure 2. Illustration of ILBP value generation for a 3×3 gray image window, where $M = 8$, Radius = 1 and $I_{mean} = 94$. The bit representing the center pixel has been underlined in the binary representation of the LBP value.

2.2. Rotation Invariant LBP (RILBP)

RILBP [22] is achieved by bit-wise rotation (circularly) of the binary patterns and then by selecting the minimum value. This is done to cancel out the effect of rotation on a texture, which changes the pattern, although the texture in consideration is essentially the same. RILBP can formally be defined as follows:

$$RILBP_{(M,R)}(x_{cen}, y_{cen}) = \min\{Rot(LBP_{(M,R)}, i) | 0 \leq i \leq M - 1\}. \quad (6)$$

Here, $Rot(A, i)$ is a function that takes an M-bit binary pattern 'A' and performs i time circular bit-wise right shift operation on 'A'. The entire process is shown in Figure 3.

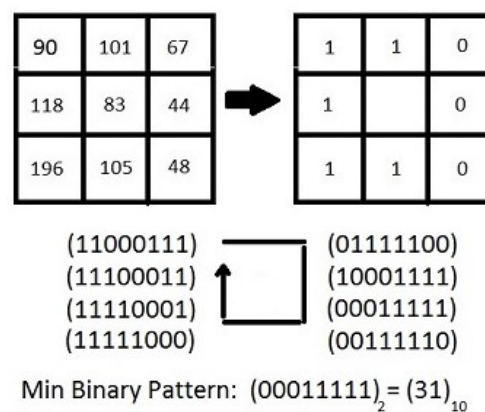


Figure 3. Illustration of RILBP value generation for a 3×3 gray image window, where $M = 8$ and Radius = 1. The binary pattern is rotated clockwise here.

2.3. Uniform LBP (ULBP)

In ULBP [22], the binary patterns with less than or equal to two numbers of zero/one transitions are considered as uniform patterns and the rest are considered as non-uniform patterns. In this variant

of LBP, all the non-uniform patterns are marked with the same label, whereas, for uniform patterns, different labels are used, one for each pattern. This is performed because it has been observed that certain patterns constitute a major portion of all texture features. ULBP uses, $M \times (M - 1) + 3$ symbols to label the patterns.

2.4. Rotation Invariant and Uniform LBP (RIULBP)

In RIULBP [22], the patterns are chosen such that they are both rotation invariant and uniform. Similar to ULBP, here also all non-uniform rotation invariant patterns are placed in one separate bin. This variant of LBP can be formulated as

$$RIULBP_{(M,R)}(x_{cen}, y_{cen}) = \begin{cases} \sum_{n=1}^M f(I_n - I_{cen}), & \text{if } U(RILBP_{(M,R)}(x_{cen}, y_{cen})) \geq 2, \\ M + 1, & \text{otherwise.} \end{cases} \quad (7)$$

Here,

$$U(RILBP_{(M,R)}(x_{cen}, y_{cen})) = \left(\sum_{n=2}^M |f(I_n - I_{cen}) - f(I_{n-1} - I_{cen})| + |f(I_M - I_{cen}) - f(I_1 - I_{cen})| \right). \quad (8)$$

2.5. Robust and Uniform LBP (RULBP)

In the present work, we have proposed a minor but significant modification to Robust LBP (RLBP) [24] to develop RULBP. In RLBP, the argument of the function $f(x)$ i.e., $(I_n - I_{cen})$ (see Equation (2)) is replaced with $(I_n - I_{cen} - th)$, where th acts as a threshold value. This essentially means that the value of I_n has to be greater than the center pixel's gray value I_{cen} by an amount th to produce a 1 (see Figure 4). This descriptor is devised with the idea of increasing the robustness to negligible changes in gray value. Therefore, the RLBP can be formally defined as follows:

$$RLBP_{(M,R)}(x_{cen}, y_{cen}) = \sum_{n=1}^M f(I_n - I_{cen} - th) \times 2^{n-1}. \quad (9)$$

In this work, we have given a notion of setting the value of th for text/non-text separation in handwritten documents and also incorporated the idea of 'uniform pattern' in RLBP to develop RULBP.

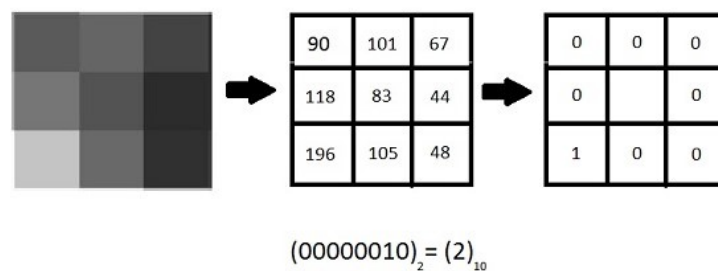


Figure 4. Illustration of RLBP value generation for a 3×3 gray image window, where $M = 8$ and Radius = 1. Here, the value of $th = 90$.

2.5.1. Idea of 'Uniform Pattern'

To prove the effectiveness of LBP for texture classification [22], it has been shown that over 90 percent of the LBPs (generated using a segment of the image) present in a textured surface are 'uniform patterns'. Besides that, as 'uniform patterns' consider a very limited number of 0/1 transition, they can efficiently detect the common microfeatures like corner, edge and spots. Thus,

in the present work, we have amalgamated the concept of ‘uniform patterns’ with RLBP to generate RULBP. The formal definition of RULBP is given below:

$$RULBP_{(M,R)}(x_{cen}, y_{cen}) = \begin{cases} \sum_{n=1}^M f(I_n - I_{cen} - th), & \text{if } U(RILBP_{(M,R)}(x_{cen}, y_{cen})) \geq 2, \\ M + 1, & \text{otherwise.} \end{cases} \quad (10)$$

The value of $U(RLBP_{(M,R)}(x_{cen}, y_{cen}))$ is computed using Equation (8).

2.5.2. Selecting the Value of th

From Equation (9), it can be inferred that the threshold (th) in RLBP plays an important role and whose value might be application specific to some extent. Thus, in this work, we have attempted to rationalize it in the context of text/non-text separation in handwritten documents.

Most handwritten documents generally possess a large intensity variation at the stroke level due to the varied nature of writing instruments and non-uniformity in the amount of pressure applied while writing. This non-homogeneity over a single stroke can only be identified if we magnify the image (see the dark and bright patches within the stroke in Figure 5. For example, LBP for the 3×3 segment, marked in red, in Figure 5 is ‘00010001’. However, the visual perception of a human being considers this as a homogeneous region with all zeros ‘00000000’. This property of handwritten documents may generate erroneous LBP feature values, which, in turn, fail to distinguish the text components from the non-text ones. In order to solve such problems, a threshold ‘ th ’ has been introduced in LBP to generate RLBP. This threshold ensures that two gray values that are not perceptibly different are not labeled differently. The problem with selecting a value of th is that, if the value is extremely large, then the entire region will behave like a homogeneous region with no intensity variation. This is because the binary pattern, according to Equation (10), will be all zeros for every pixel. Therefore, we need to provide an upper limit, th_{max} , on the value of th .

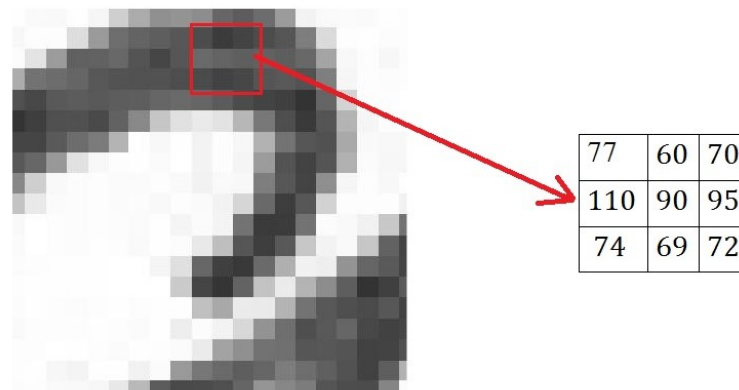


Figure 5. Magnified image of a stroke shows the variation in gray values. A 3×3 matrix shows the intensity values of the gray image segment marked in red.

To address this issue, we have set an upper limit, th_{max} , on the value of th . Generally, in a real-life handwritten document image, the intensity of the background pixels reside within a close proximity of the maximum intensity 255. Here, we assume that the intensity of the background pixels will be in a range of $[245, 255]$. Now, for each image, we find the highest gray-scale intensity ($I_{graymax}$) less than 245. We claim that the pixel P having this intensity value has to be a part of some writing stroke. th_{max} has to be such that, if we consider I_{cen} has a value $I_{graymax}$ and a neighboring pixel has a value 245, $f(x)$ as given in Equation (2) for $x = I_n - I_{cen} - th$ gives a value 1. Therefore, $th_{max} = 245 - I_{graymax}$. The value of th can be anything between th_{max} and 0. We have performed a weighted average of the threshold values in the range, with the weights increasing for higher values of th and found the ideal

threshold th_{ideal} to be at around a value of 100. We have taken various threshold values from 5 to 115 and found experimentally that the accuracy of classification is maximum at about a threshold of 100. It is to be noted that we have set this hardcore threshold value after conducting an exhaustive experimentation on the images belonging to our dataset. A change in document images might change the threshold value a bit, but, we foretell that, this assumption would give the researchers a clear hint to set the threshold value for the document images they consider.

3. Method

The input color image is first converted to the grayscale image and then the connected components (CCs) are extracted for feature computation and classification. The entire process is depicted in Figure 6. For CC extraction, first the grayscale image is binarized and the bounding boxes (BBs) of all of the eight-connected components in the binarized image are calculated. Then, using these estimated bounding boxes, CCs from the corresponding grayscale image are extracted. As we are considering real-world handwritten documents, we need to be very careful about the noise present in these documents, which might affect the binarization and BB estimation process. Thus, for effective binarization, a background estimation and separation procedure is followed, prior to the actual binarization, using Otsu's method as given in [27]. During BB estimation from the binarized image, only the CCs having height and width greater than three pixels are considered to avoid noise. After extraction of the CCs from the grayscale image, six different LBP based features are computed. During feature computation, the radius R has been kept constant at 1 (i.e., the number of neighboring pixels $M = 8$). In order to compute a feature vector for each CC, we have generated a normalized histogram of those LBP values. The number of bins used depends on the particular LBP variant considered. Here, we should also point out that the LBP operators have been applied to each and every pixel of a CC, without any discrimination.

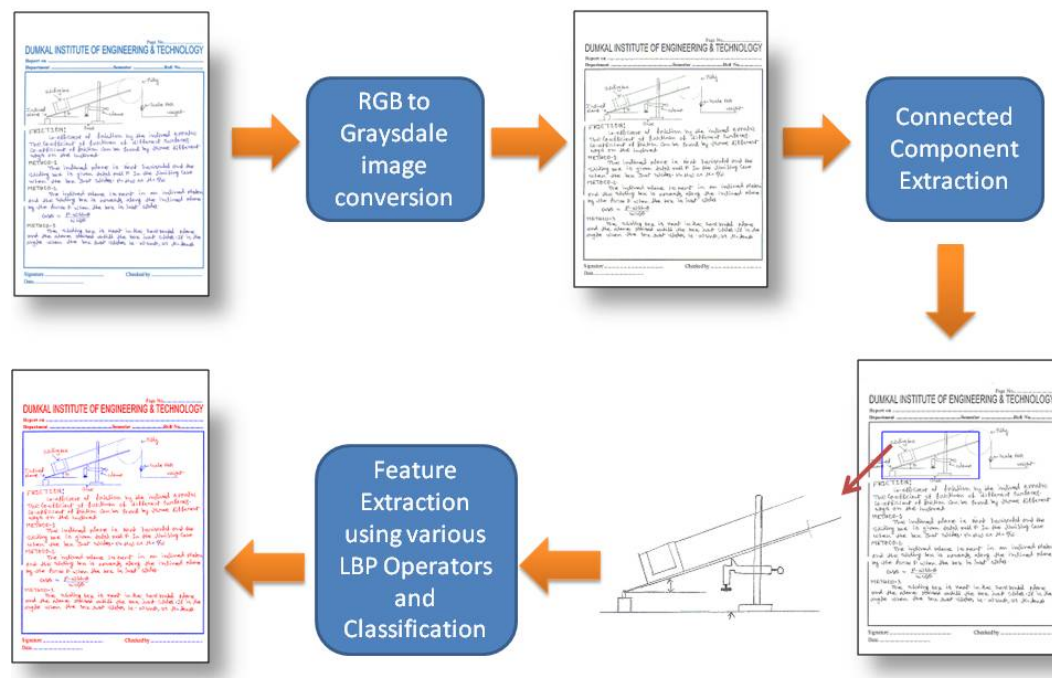


Figure 6. Flowchart of the entire text/non-text separation process.

4. Experimental Setup

Experimental setup for any pattern classification problem requires an annotated dataset, classifiers and a set of evaluation metrics. In this section, the data preparation procedure is described first,

followed by details of the parameter values used by the classifiers. At the end, we present the evaluation metrics used in the experiment.

4.1. Database Preparation

It is found that the unavailability of a standard database may be one of the possible reasons for slow progress in some research areas, such as text/non-text separation from handwritten documents in spite of their importance. Keeping this fact in mind, in the present work, a database has been developed that consists of 104 handwritten engineering lab copies and class notes collected from an engineering college. These copies include textual contents along with a varying number of tables, graphic components and some printed texts. All of these lab copies are written by more than 20 students from different engineering and science streams. The age of the writers vary from 18 to 24. Please note that all these copies are written either in English or Bangla. The collected documents are scanned in 300 DPI (Dots per inch) using a flatbed scanner and then these scanned copies are stored as 24 bit 'BMP' files. A sample image from the current database is shown in Figure 7a and the corresponding ground truth image is shown in Figure 7b. In this work, from those 104 handwritten pages, a total of 66,058 CCs are extracted, out of which 25,011 are text components and 41,047 are non-text components.

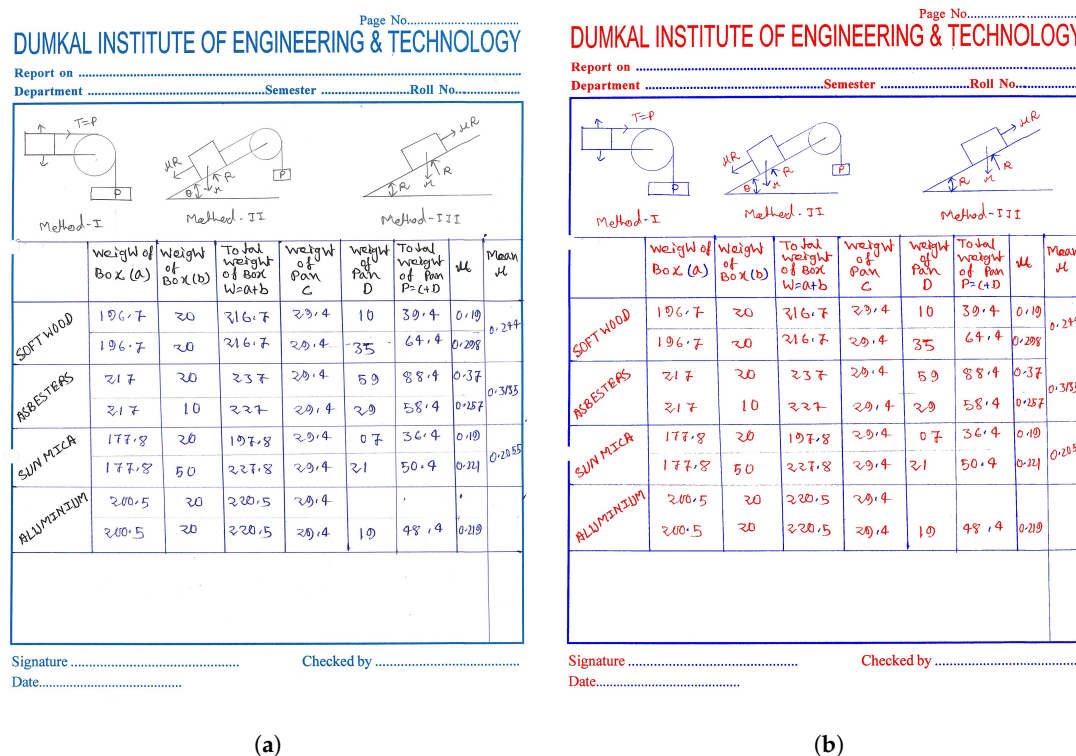


Figure 7. (a) sample image from our dataset; (b) ground truth of the given image (here, red represents text and blue represents non-text components).

4.2. Classifiers

For classification of the extracted CCs, five well-known classifiers are used in this work, namely, Naïve Bayes (NB), Multi-layer perceptron (MLP), K-nearest neighbor (K-NN), Random forest (RF) and Support Vector Machine (SVM). In the current experimental setup, performances of Simple LBP, ILBP, RILBP, ULBP and RIULBP descriptors with each of the considered classifiers for the present dataset are measured. Then, the classifier that performs better in all or most cases is used to justify the newly hypothesized 'uniform pattern' in RLBP i.e., RULBP. It is to be noted that one of the key parameters of RULBP is th whose value is subjective to the document image. Here, different trial runs

are performed to choose the optimal value of th . In this work, Weka 3 [28], a data mining software (University of Waikato, Hamilton, New Zealand), has been used for classification and visualization purpose. The values of the classifiers' parameters used in the current experiment are given in Table 1.

Table 1. Detail values of the parameters used by the classifiers under consideration.

Classifier	Parameters with Values
NB	<ul style="list-style-type: none"> • Batch size: 100 • Normal distribution for numeric attributes
MLP	<ul style="list-style-type: none"> • Learning Rate for the back propagation algorithm: 0.3 • Momentum Rate: 0.2 • Number of epochs to train through: 500 • Learning Rate: 0.3
SMO	<ul style="list-style-type: none"> • Complexity constant C: 1 • Tolerance Parameter: 1.0×10^{-3} • Epsilon for round-off error: 1.0×10^{-12} • The random number seed: 1
K-NN	<ul style="list-style-type: none"> • K: 1 • Batch size: 100
RF	<ul style="list-style-type: none"> • Batch size: 100 • Minimum number of instances per leaf: 1 • Minimum numeric class variance proportion of train variance for split: 1.0×10^{-3} • The maximum depth of the tree: unlimited

4.3. Performance Metrics

The performances of the LBP variants are measured using the following conventional metrics:

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

$$FM = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (13)$$

$$Accuracy = \frac{TP + TN}{Total\ number\ of\ samples} \times 100\%. \quad (14)$$

In Equations (11)–(14), TP , FP , TN and FN represent true positive, false positive, true negative and false negative, respectively. It is to be noted that all the experiments are done using 3-fold cross validation and the final results are computed after taking the average performance of the three folds.

5. Experimental Results

Detailed results for each LBP based feature descriptors except RULBP with each of the five classifiers for the current database are given in Table 2. From Table 2, it can be observed that the RF classifier outperforms others. Thus, classification results for RULBP with different threshold values are computed using RF classifier only. We also see that the RULBP operator gives the best accuracy in classification, among all the LBP variants considered. Detailed results depicting the performance of RULBP for different thresholds are given in Table 3. A pictorial comparison among the performances of different LBP operators using RF classifier is given in Figure 8. Figure 9 shows the image of a document containing text written in Bangla and classified using RULBP, which gives the best result among all LBP variants. In addition to this, a graphical comparison of the performance of various LBP

variants are also presented in Figures 10 and 11. The data in Table 2 forms the basis for the points in Figure 10 while the data in Table 3 forms the basis for the points in Figure 11.

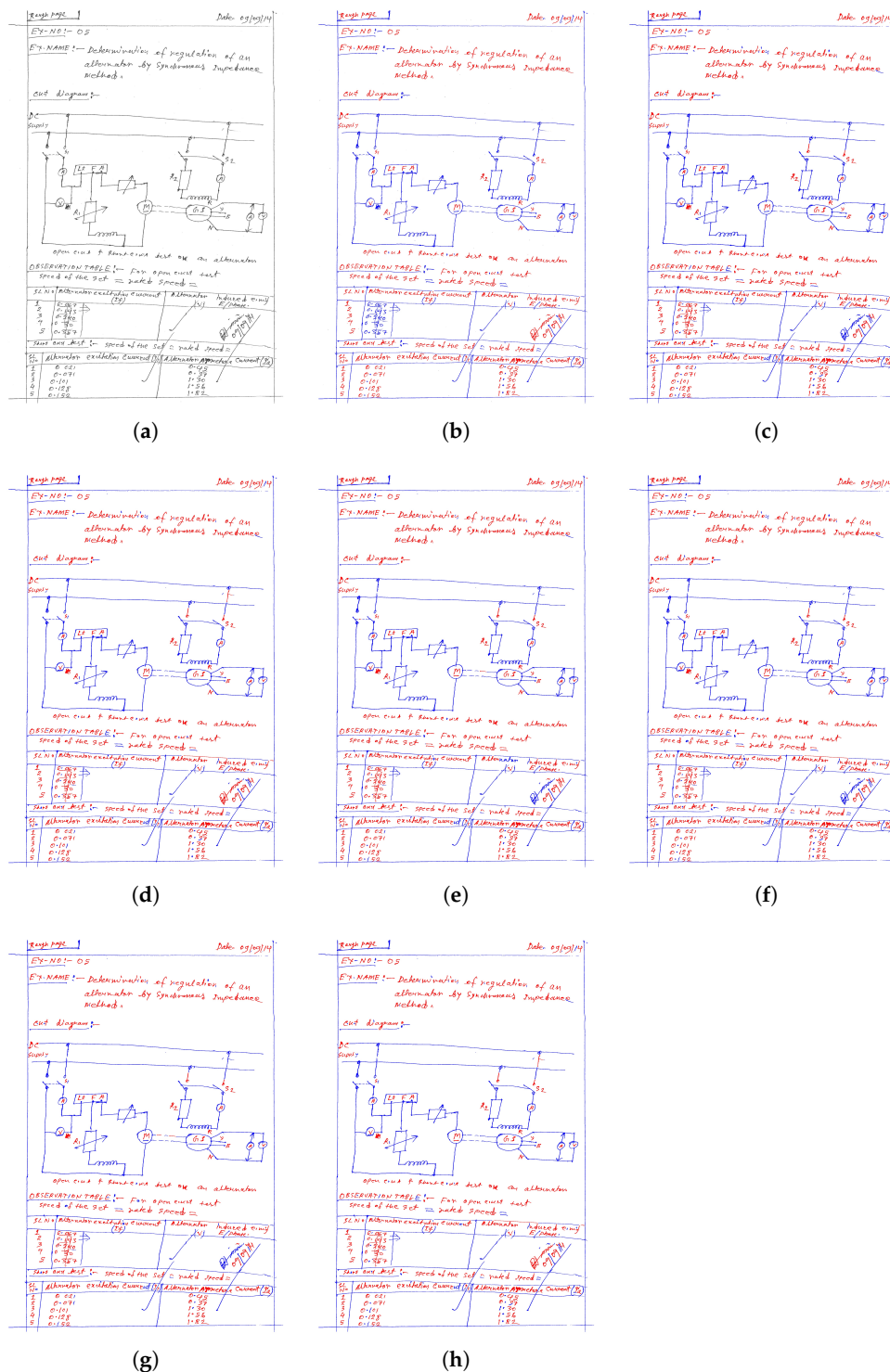


Figure 8. Pictorial comparison between the performances of different LBP based features with RF Classifier. Here, (a) grayscale image, (b) ground truth image, (c) result using LBP, (d) result using ILBP, (e) result using RILBP, (f) result using ULBP, (g) result using RIULBP, and (h) result using RULBP.

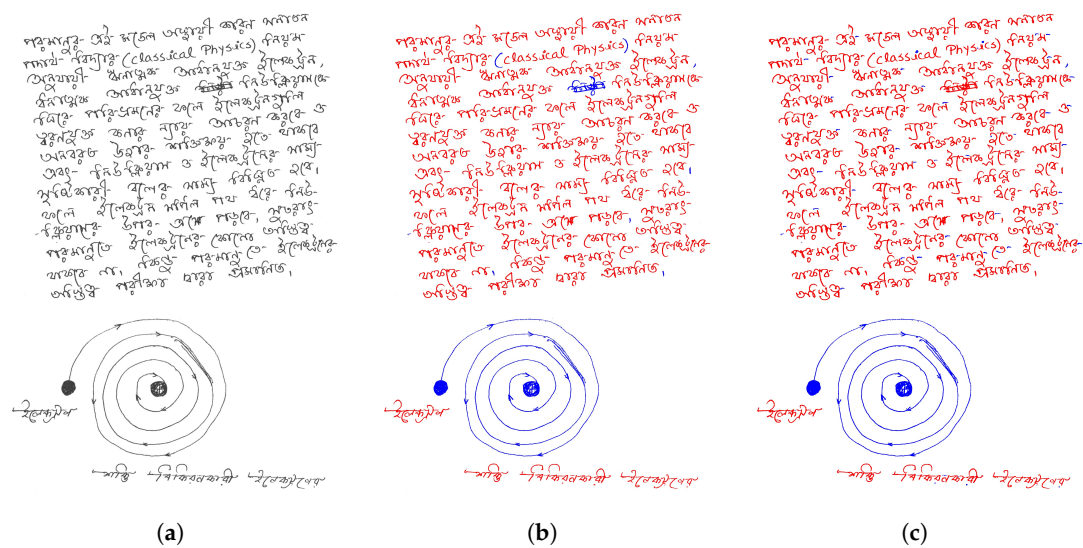


Figure 9. A Bangla handwritten document classified with RF classifier. Here, (a) grayscale image, (b) ground truth image, (c) result using RULBP.

Table 2. Performance measure for text/non-text separation, using various LBP features.

Feature	Feature Dimension	Classifier	Precision	Recall	F-Measure	Accuracy (in %)
LBP	256	NB	0.802	0.771	0.774	77.08
		MLP	0.529	0.54	0.534	54.04
		SMO	0.892	0.889	0.889	88.87
		K-NN	0.856	0.851	0.852	85.12
		RF	0.914	0.913	0.913	91.33
ILBP	511	NB	0.82	0.764	0.767	76.41
		MLP	0.386	0.621	0.476	62.13
		SMO	0.862	0.858	0.859	85.84
		K-NN	0.852	0.845	0.847	84.5
		RF	0.913	0.913	0.912	91.31
RILBP	36	NB	0.831	0.802	0.805	80.18
		MLP	0.908	0.907	0.905	90.66
		SMO	0.889	0.886	0.887	88.62
		K-NN	0.882	0.882	0.882	88.19
		RF	0.912	0.912	0.912	91.23
ULBP	59	NB	0.862	0.857	0.858	85.65
		MLP	0.912	0.912	0.912	91.22
		SMO	0.891	0.888	0.889	88.8
		KNN	0.901	0.901	0.901	90.13
		RF	0.918	0.918	0.917	91.79
RIULBP	10	NB	0.859	0.855	0.856	85.52
		MLP	0.907	0.907	0.906	90.71
		SMO	0.888	0.886	0.887	88.58
		KNN	0.886	0.886	0.886	88.62
		RF	0.909	0.909	0.908	90.9

Table 3. Classification results using various thresholds for RULBP. The classification accuracy gradually increases and attains a maximum at a *th* of 105 units.

Feature Dimension	Threshold (<i>th</i>)	Precision	Recall	F-Measure	Accuracy in %
59	5	0.915	0.915	0.914	91.45
	25	0.915	0.915	0.915	91.52
	45	0.916	0.916	0.915	91.61
	65	0.917	0.917	0.917	91.72
	85	0.919	0.918	0.918	91.84
	105	0.920	0.920	0.919	91.96
	115	0.919	0.918	0.918	91.84

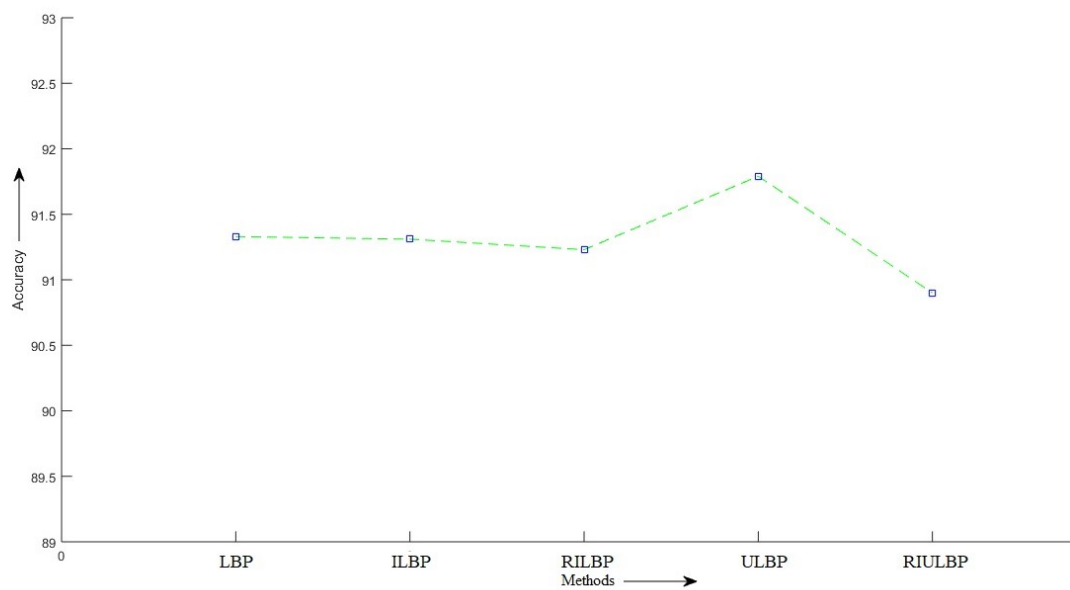


Figure 10. Graphical comparison of the performances of different LBP variants in classifying the texts and non-texts present in handwritten document images.

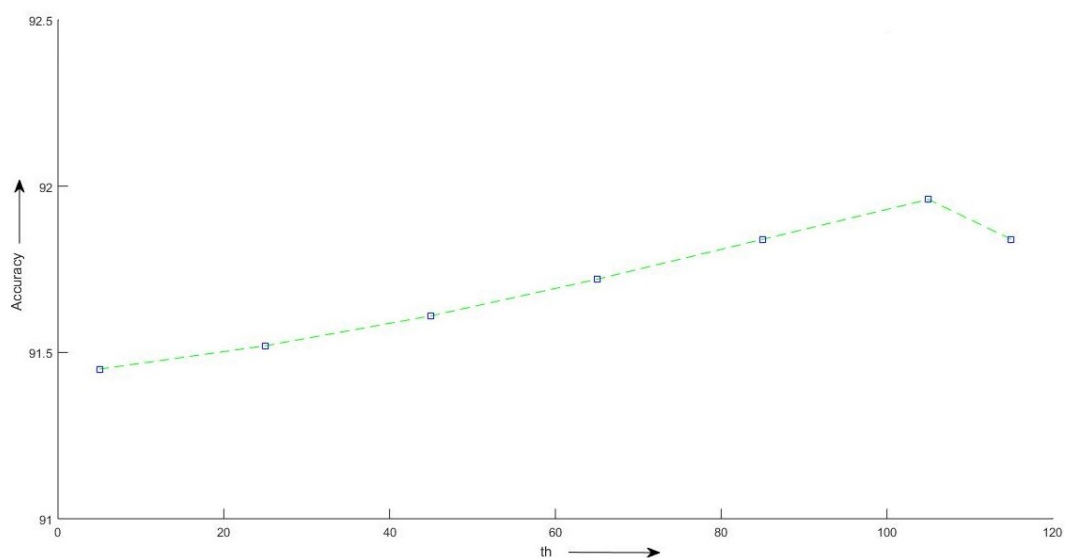


Figure 11. Graphical comparison of the performances of RULBP with different threshold in classifying the texts and non-texts present in handwritten document images.

In the literature, different texture feature descriptors have been used to separate text and non-text regions in printed documents. Here, we have considered two of the recent ones and compared their individual performances on our dataset, with the performance of the RULBP operator. One of the methods uses GLCM as feature descriptor [4] while the other uses Histogram of Oriented Gradients (HOG) [29]. Table 4 gives the accuracy of classification for each of the three feature descriptors using all five classifiers. It can be seen that the RULBP operator outperforms the other feature descriptors in most cases.

Table 4. Performance comparison in terms of recognition accuracy (in %) of GLCM, HOG and RULBP ($th = 105$) on the present dataset for five different classifiers.

Method	NB	MLP	SMO	KNN	RF
RULBP	50.38	90.78	88.62	90.20	91.96
GLCM	77.92	90.22	87.21	87.70	90.90
HOG	36.22	80.46	72.61	88.89	91.42

6. Conclusions

In the present work, our objective is to validate the utility of LBP based feature descriptors for the classification of text and non-text components present in handwritten documents, in a comprehensive way. We have experimentally shown that RLBP performs better than simple LBP, ILBP, RILBP, ULBP and RIULBP. However, a major issue in using RLBP is the selection of a suitable threshold, which might be domain specific. In the current research attempt, we have selected the optimal value of the threshold on the basis of a few observations, which is also validated through an experiment. We have provided a justification for this selection as well, which we believe would lead to deeper insight into the selection of the threshold used for LBP, especially in the case of handwritten documents. Excluding that, we have proposed a minor modification to RLBP by incorporating the concept of a ‘uniform pattern’ to develop RULBP, and it has been shown experimentally that RULBP performs better than RLBP. In the future, we would look for the other texture based features along with some other variants of LBP to see their utility in the current context. In the future, we plan to enlarge the database by incorporating various types of document images, which, in turn, would motivate more researchers to do some tangible work. It is worth mentioning here that, in order to analyze the texts written in different scripts, a script recognition module is required [30], since an OCR engine is script specific. Thus, our future plan is to incorporate the same in our model to make it more useful in a multi-script environment. Another area that we will look into is the generalization of the threshold value th , so that we may formulate a solid set of procedures that can be useful for any document, instead of using an empirical method to detect the same.

Acknowledgments: This work is partially supported by the Center for Microprocessor Application for Training Education and Research (CMATER) research laboratory of the Computer Science and Engineering Department, Jadavpur University, India, and PURSE-II and UPE-II Jadavpur University projects. Ram Sarkar is partially funded by a DST grant (EMR/2016/007213).

Author Contributions: The first three authors—Sourav Ghosh, Dibyadwati Lahiri and Showmik Bhowmik—have contributed equally to the paper. Ergina Kavallieratou and Ram Sarkar provided essential guidance and corrections at various stages of the work.

Conflicts of Interest: The authors have no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LBP	Local Binary Pattern
GLCM	Gray-Level Co-Occurrence Matrix
CC	Connected Components
BB	Bounding Box

ILBP	Improved Local Binary Pattern
RILBP	Rotation Invariant Local Binary Pattern
ULBP	Uniform Local Binary Pattern
RIULBP	Rotation Invariant Uniform Local Binary Pattern
RULBP	Robust Uniform Local Binary Pattern
NB	Naive Bayes
MLP	Multilayer Perceptron
SMO	Sequential Minimal Optimization
k-NN	k-Nearest Neighbors
RF	Random Forest

References

1. Santosh, K.C.; Wendling, L. Character recognition based on non-linear multi-projection profiles measure. *Front. Comput. Sci.* **2015**, *9*, 678–690.
2. Santosh, K.C.; Iwata, E. Stroke-Based Cursive Character Recognition. In *Advances in Character Recognition*; InTechOpen: London, UK, 2012; Chapter 10.
3. Santosh, K.C.; Nattee, C.; Lamiroy, B. Relative Positioning Of Stroke-based Clustering: A New Approach To Online Handwritten Devnagari Character Recognition. *Int. J. Image Graph.* **2012**, *12*, 1250016, doi:10.1142/S0219467812500167.
4. Oyedotun, O.K.; Khashman, A. Document segmentation using textural features summarization and feedforward neural network. *Appl. Intell.* **2016**, *45*, 198–212.
5. Le, V.P.; Nayef, N.; Visani, M.; Ogier, J.-M.; De Tran, C. Text and non-text segmentation based on connected component features. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; Volume 45, pp. 1096–1100.
6. Tran, T.-A.; Na, I.-S.; Kim, S.-H. Separation of Text and Non-text in Document Layout Analysis using a Recursive Filter. *KSII Trans. Internet Inf. Syst.* **2015**, *9*, 4072–4091.
7. Sarkar, R.; Moulik, S.; Das, N.; Basu, S.; Nasipuri, M.; Kundu, M. Suppression of non-text components in handwritten document images. In Proceedings of the 2011 International Conference on Image Information Processing (ICIIP), Shimla, India, 3–5 November 2011; pp. 1–7.
8. Bhowmik, S.; Sarkar, R.; Nasipuri, M. Text and Non-text Separation in Handwritten Document Images Using Local Binary Pattern Operator. In *International Conference on Intelligent Computing and Communication*; Springer: Singapore, 2017; pp. 507–515.
9. Santosh, K.C. g-DICE: Graph mining-based document information content exploitation. *Int. J. Doc. Anal. Recognit. IJDAR* **2015**, *18*, 337–335, doi:10.1007/s10032-015-0253-z.
10. Santosh, K.C. Complex and Composite Graphical Symbol Recognition and Retrieval: A Quick Review. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*; Springer: Singapore, 2016.
11. Vil'kin, A.M.; Safonov, I.V.; Egorova, M.A. Algorithm for segmentation of documents based on texture features. *Pattern Recognit. Image Anal.* **2013**, *23*, 153–159.
12. Park, H.C.; Ok, S.Y.; Cho, H. Word extraction in text/graphic mixed image using 3-dimensional graph model. In Proceedings of the ICCPOL, Tokushima, Japan, 24–26 March 1999; Volume 99, pp. 171–176.
13. Shih, F.Y.; Chen, S.-S. Adaptive document block segmentation and classification. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 797–802.
14. Antonacopoulos, A.; Ritchings, T.R.; De Tran, C. Representation and classification of complex-shaped printed regions using white tiles. In Proceedings of the Third International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 2, pp. 1132–1135.
15. Pintus, R.; Yang, Y.; Rushmeier, H. ATHENA: Automatic text height extraction for the analysis of text lines in old handwritten manuscripts. *J. Comput. Cult. Herit.* **2015**, *8*, 1.
16. Yang, Y.; Pintus, R.; Gobbetti, E.; Rushmeier, H. Automatic single page-based algorithms for medieval manuscript analysis. *J. Comput. Cult. Herit.* **2017**, *10*, 9.

17. Garz, A.; Sablatnig, R.; Diem, M. Layout analysis for historical manuscripts using sift features Document. In Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR), Beijing, China, 18–21 September 2011.
18. Garz, A.; Sablatnig, R.; Diem, M. Using Local Features for Efficient Layout Analysis of Ancient Manuscripts. In Proceedings of the European Signal Processing Conference, Barcelona, Spain, 29 August–2 September 2011; pp. 1259–1263.
19. Wang, D.; Nihari, S.N. Classification of newspaper image blocks using texture analysis. *Comput. Vis. Graph. Image Process* **1989**, *47*, 327–352.
20. Belaïd, A.; Santosh, K.C.; d’Andecy, V.P. Handwritten and Printed Text Separation in Real Document. *arXiv* **2013**, arXiv:1303.4614.
21. Nanni, L.; Lumini, A.; Brahnam, S. Survey on LBP based texture descriptors for image classification. *Expert Syst. Appl.* **2012**, *39*, 3634–3641.
22. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987.
23. Jin, H.; Liu, Q.; Lu, H.; Tong, X. Face detection using improved LBP under Bayesian framework. In Proceedings of the Third International Conference on Image and Graphics (ICIG), Hong Kong, China, 18–20 December 2004; pp. 306–309.
24. Heikkila, M.; Pietikainen, M. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 657–662.
25. Harwood, D.; Ojala, T.; Pietikainen, M.; Kelman, S.; Davis, L. Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions. *Pattern Recognit. Lett.* **1995**, *16*, 1–10.
26. Ojala, T.; Pietikainen, M.; Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **1996**, *29*, 51–59.
27. Das, B.; Bhowmik, S.; Saha, A.; Sarkar, R. An Adaptive Foreground-Background Separation Method for Effective Binarization of Document Images. In *Eighth International Conference on Soft Computing and Pattern Recognition*; Springer: Cham, Switzerland, 2016.
28. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. The WEKA Workbench. Online Appendix for *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed.; Morgan Kaufmann: Burlington, MA, USA, 2016.
29. Sah, K.A.; Bhowmik, S.; Malakar, S.; Sarkar, R.; Kavallieratou, E.; Vasilopoulos, N. Text and non-text recognition using modified HOG descriptor. In Proceedings of the IEEE Calcutta Conference (CALCON), Kolkata, India, 2–3 December 2017; doi:10.1109/CALCON.2017.8280697
30. Obaidullah, S.M.; Santosh, K.C.; Halder, C.; Das, N.; Roy, K. Automatic Indic script identification from handwritten documents: Page, block, line and word-level approach. *Int. J. Mach. Learn. Cyber* **2017**, 1–20, doi:10.1007/s13042-017-0702-8

