01 Mar 2010

Historical Document Enhancement Using LUT Classification

Tayo Obafemi-Ajayi
Missouri University of Science and Technology, towd2@mst.edu

Gady Agam

Ophir Frieder

Recommended Citation

T. Obafemi-Ajayi et al., "Historical Document Enhancement Using LUT Classification," *International Journal on Document Analysis and Recognition*, vol. 13, no. 1, pp. 1 - 17, Springer, Mar 2010.
The definitive version is available at https://doi.org/10.1007/s10032-009-0099-3

IJDAR (2010) 13:1–17
DOI 10.1007/s10032-009-0099-3

ORIGINAL PAPER

Historical document enhancement using LUT classification

Tayo Obafemi-Ajayi · Gady Agam · Ophir Frieder

Received: 21 May 2008 / Revised: 16 July 2009 / Accepted: 9 October 2009 / Published online: 17 November 2009

Abstract The fast evolution of scanning and computing technologies in recent years has led to the creation of large collections of scanned historical documents. It is almost always the case that these scanned documents suffer from some form of degradation. Large degradations make documents hard to read and substantially deteriorate the performance of automated document processing systems. Enhancement of degraded document images is normally performed assuming global degradation models. When the degradation is large, global degradation models do not perform well. In contrast, we propose to learn local degradation models and use them in enhancing degraded document images. Using a semi-automated enhancement system, we have labeled a subset of the Frieder diaries collection (The diaries of Rabbi Dr. Avraham Abba Frieder. http://ir.iit.edu/collections/). This labeled subset was then used to train classifiers based on lookup tables in conjunction with the approximated nearest neighbor algorithm. The resulting algorithm is highly efficient and effective. Experimental evaluation results are provided using the Frieder diaries collection (The diaries of Rabbi Dr. Avraham Abba Frieder. http://ir.iit.edu/collections/).

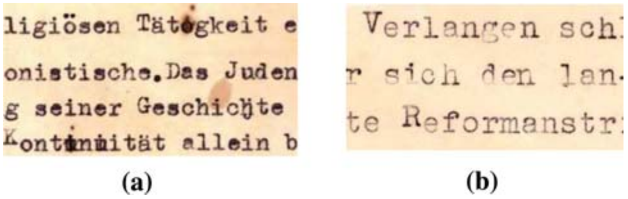
1 Introduction

Historical document collections are often very poor in quality and suffer from some form of degradation. Many of these documents have deteriorated due to the age of paper and ink used. They currently exist electronically as scanned document images. There is a significant need to convert collections of degraded historical documents into digital archives and libraries. However, large degradations make these documents hard to read and substantially deteriorate the performance of automated document processing systems. It is essential for the degradations in these document images to be corrected to facilitate their conversion to indexable digital libraries. Historical documents could be either handwritten, machine printed (typewritten) or both. We focus on the enhancement of typewritten historical documents.

Antonacopoulos et al. expound on the unique challenges facing enhancement of typewritten documents in [3,4,6]. Not only is the quality of typewritten text poor and non-uniform, but also these documents contain noisy background, paper discoloration, creases, blurred, merged and faint text [6]. Typewritten text may contain non-uniform characters, some darker or fainter than others, because each character is produced individually depending on the amount of force used in striking the typewriter keys [4] while some of the characters may be blotted (such as ‘e’s), as illustrated in Fig. 1. The degradation of the text hinders the readability of these documents, as seen in Fig. 1. The level and type of degradation varies between documents. Thus, there is need for an adaptable automated system to correct such degradations to produce enhanced document images that result in improved legibility.

Existing state-of-the-art document enhancement systems for processing historical documents focus primarily on segmentation techniques based on foreground–background separation. The text in the documents is classified as foreground while everything else is rendered as background. While such systems normally perform well in obtaining a relatively uniform background, they are unable to effectively correct distortions in the foreground such as blotted text, broken characters or overwritten characters. Often text in the

T. Obafemi-Ajayi (✉) · G. Agam · O. Frieder
Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA
e-mail: ihimtay@gmail.com



ligiösen Tätigkeit e
onistische.Das Juden
g seiner Geschichte
Kontinnüität allein b

Verlangen schl
r sich den lan-
te Reformanstr

Fig. 1 Different degradations in typewritten documents. **a** Blotted/filled characters, **b** fainted characters

document is further degraded during the foreground–background separation. Our proposed approach improves on current state of the art systems in its ability to correct text degradations in typewritten documents, beyond foreground–background separation.

We present an automated adaptive system, based on lookup table (LUT) classification algorithms, which learns the corrections of patterns of text degradation in document images. In contrast to known classifiers such as the Support Vector Machine (SVM) classifier [14], our LUT classifier is simple and efficient design and so can handle large descriptors and inherent ambiguities. We use actual degraded historical documents for learning a degradation model using a subset of the Frieder document collection [17]. Our expert-labeled data, the ground truth document images, are generated by a human expert using an interactive document enhancement software [10]. The software allows a human expert to manually correct the degradations in a document character by character to generate an ideal clean text document image of the degraded document, as illustrated in Fig. 2, which constitutes our expert-labeled data. We evaluate the performance of our system by applying it to a different set of test data obtained from the same collection. The performance of our system is measured both quantitatively (Misclassification Error [27]) and qualitatively (enhanced readability) in comparison with the ground truth data. Our automated system is substantially more efficient for correcting a large set of documents compared with manual processing.

The main contributions of this paper are in the development of LUT classifiers, which effectively learn patterns of degradations and their correction directly from the data and in the design of a system that can efficiently process multiple documents. Our proposed method is general, in that the degradation model is directly learned from the actual degraded images and their corresponding ground truth images.

This paper improves on the preliminary results in [25], in that we perform a more extensive and accurate conceptual and experimental evaluation of our work. In addition, we include an extensive and very thorough comparison of the proposed approach to current state-of-the-art methods in terms of quantitative and qualitative measures. The comparisons discussed in this work demonstrate that the proposed approach outperforms known state-of-the-art enhancement techniques and a very strong general purpose machine learning algorithm (SVM).

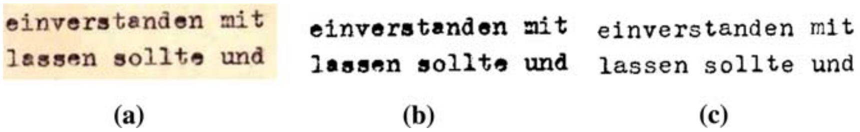
The paper is organized as follows: we discuss related approaches in Sect. 2. The proposed LUT classifier is described in Sect. 3. Section 4 provides both qualitative and quantitative experimental evaluation of the proposed approach. Section 5 concludes the paper.

2 Related work

Existing foreground–background separation-based systems for enhancing degraded historical documents include the work done by Gatos et al. in [20] and Agam et al. in [1]. The system developed by Gatos et al. handles binarization of historical documents using an adaptive threshold segmentation and various pre- and post-processing steps. An iterative approach for segmenting degraded document images is described by Kavallieratou et al. [22]. The work done by Agam et al. is based on probabilistic models utilizing the expectation maximization (EM) algorithm.

Antonacoupoulos et al. [6] propose a method to convert historical documents to a logically indexed, searchable form. Their approach is based on content extraction using semantic information that involves the expert knowledge of a historian/archivist. Antonacoupoulos et al. in [5] attempt to enhance these documents to prepare them for optimal OCR performance using an off-shelf OCR package. They attempt to enhance the documents by individually segmenting and enhancing each character.

Allier et al. in [2] propose an approach for handling restoration of character shapes in antique document images based on Gabor filtering and active contours model. Their 'historical' character restoration method is aimed at preserving the original shape of the observed characters by restoration of broken patches in the individual character images. The work by Cannon et al. [13] deals with different types of image



einverstanden mit
lassen sollte und

einverstanden mit
lassen sollte und

einverstanden mit
lassen sollte und

Fig. 2 Example of the segmented (binary) image and the ground truth data derived from an original degraded document image using the interactive document enhancement software. **a** Original degraded image, **b** segmented degraded image, **c** ground truth image (by a human expert)

degradations in a typewritten document archive, such as touching characters, broken characters and salt-and-pepper noise. They quantify document image quality for predicting OCR accuracy. Based on the information about the quality of the image, they train a linear classifier that will predict the best restoration method. However, their quality measures do not take into account large degradations such as blotted characters found in degraded typewritten document images. Their method is OCR dependent as the process of training the linear classifier is based on the OCR accuracy feedback received about the input document images.

An adaptive approach to text image restoration using feed-forward neural networks based on multilayer perceptrons (MLPs) is proposed in [29]. Using the output from an OCR system and a distorted text image, they train an adaptive restoration filter and then apply the filter to the distorted text image that the OCR system could not recognize. MLPs are used as filters with a square input window to model the inverse of the distortion process. This allows one to adapt the filter by retraining the MLP on each separate page to be processed. However, the filters must be applied under human guidance to avoid further image degradation. When dealing with a large document collection, there is need for an automated system that does not rely on a human expert to select the best restoration method for each document.

Zheng et al. in [32] developed an LUT-based algorithm to restore document images using morphological degradation models. They build a lookup table, similar to our approach, using a 3 × 3 filter. However, their lookup table consists of a matrix mapping each entry to at most 512 possible outputs, unlike our approach that maps each entry to two possible outputs. In contrast to their method of correcting patch by patch, our approach corrects one pixel at a time, taking into account the neighborhood (or patch) pixel information, and so gives more accurate results. In our approach, we use actual degraded document images during training phase instead of utilizing synthetic images generated using the Kanungo morphological degradation model [31]. This degradation model is well suited for small perturbations [9] encountered during photocopying and scanning of uniform text documents, but unable to handle the large degradations found in historical typewritten documents. We discuss more extensively and compare our approach to Zheng et al.'s restoration algorithm based on Kanungo's degradation model in Sect. 4.5.

Some work have also been done specifically to enhance the binarization results of degraded images. In the work done by Gatos et al. [20], a postprocessing technique that consists of a series of shrink and swell filtering operations in the final phase of their algorithm. The purpose was to eliminate the noise obtained from segmentation, improve the quality of text regions and preserve stroke connectivity by isolated pixel removal and filling of possible breaks, gaps or holes. Mean and median filters [16], and morphological operations [30] are also usually used as a postprocessing technique to smoothen image data, thus eliminating noise. These operations usually require proper fine tuning of many parameters per document image to obtain a quality result. In contrast, as we demonstrate in Sect.4.7.2, the proposed LUT algorithm learns the pattern of degradation and correction using actual degradation models and does not involve fine tuning of multiple parameters. Moreover, the proposed approach may be applied in conjunction with other enhancement filters.

3 Ensemble LUT classification

We propose an efficient approach to enhance historical typewritten document images using effective LUT classifiers [25] that are trained to learn the patterns of degradation and correction from an expert-labeled data set. The goal of the training phase is to build the lookup table which is utilized during the correction phase to correct the degradations in the document images. In this section, we describe in detail each key process in the proposed system which include the ground truth generation 3.1, the initial binarization 3.2, the learning phase 3.3 and the correction phase 3.4.

3.1 Ground truth generation

The expert-labeled dataset consists of pairs of a binary degraded document image and the corresponding ground truth image. The ground truth data generation plays a significant role in any system that involves machine learning techniques, such as ours. As mentioned in Sect. 1, the ground truth generation for the labeled dataset is done by a human expert using the interactive document software developed by Bal et al. [10]. The software is a semi-automated editing tool which assisted by human intervention produces a high quality ground truth data of the degraded image, as shown in Fig. 2. The system performs the foreground separation, character clustering and suggests initial labeling automatically. The human expert then continues the process by labeling each character manually. The enhanced characters obtained in the final result are an average of good quality characters found in the document image, based on human judgment.

The advantage of using this software is that the quality of enhanced characters is not limited to one document image but can be averaged over the document collection. The ground truthing process requires time (approximately 3 h per document image depending on its level of degradation and the speed of the processor) and significant human input. Thus, there is a need for an automated system, such as the proposed approach, which does not rely on human intervention. However, once a degraded image has been ground truthed, it can be reused for speeding up the labeling process. In addition, as we demonstrate in Sect. 4.2.3, a small labeled sub-set of the document collection is sufficient to train the LUT (low sample complexity).

3.2 Preprocessing: initial image binarization

The LUT classifier processes binary document images consisting of only black (foreground) and white (background) pixels. Historical documents are normally stored electronically as scanned color or grayscale document images. Thus, we preprocess the document images to convert each scanned degraded document image to a binary image by separating the foreground from the background [1]. The binary image can then be effectively processed by our classifier. The preprocessing phase attempts to remove background degradations to generate a uniform background. The nature of background degradation varies from document to document. For example, some have dark streaky background while others have blots of ink stains, wrinkling, etc. [6].

There are different segmentation algorithms that can be employed to obtain a binary document image, as detailed in the survey of thresholding algorithms by Sezgin et al. [27]. Such algorithms adapt to the varied nature of background degradation in document images. We utilize the adaptive Bernsen's Min–Max threshold algorithm [11] in the segmentation process because of its known efficiency [8].

The Bernsen algorithm is based on the estimation of a local threshold value for each pixel. This value is assigned as the local threshold value only if the difference between the lowest and the highest gray-level value is bigger than a threshold L. Otherwise, it is assumed that the window region contains pixels of one class (foreground or background). The local threshold value is computed as:

$$T(x, y) = \begin{cases} \frac{I_{\text{low}}+I_{\text{high}}}{2}, & \text{if } I_{\text{high}} - I_{\text{low}} \geq L \\ I_T, & \text{if } I_{\text{high}} - I_{\text{low}} < L \end{cases} \tag{1}$$

where I_{low} and I_{high} are the lowest and the highest gray-level value in a $N \times N$ window centered at the pixel (x, y), and I_T a global threshold value.

Our LUT classifier system is general in that it does not dependent on the application of a specific foreground–background separation algorithm. Given an existing binary document image, obtained using any thresholding technique, degradations in the text in the binary image can be enhanced by feeding it directly into our system. The proposed system is designed to correct degradations by learning the correction patterns from comparison with ground truthed data. Thus, the type of method used for thresholding is of minor significance, as long as the algorithm yields a decent performance.

Intuitively, the better the thresholding method, the less degradations there are to correct. We do recommend the use of an adaptive thresholding algorithm, as such methods tend to perform relatively well on historical collections compared to global thresholding techniques. The advantage of the proposed approach is that it is a general method that learns from the degraded data it is presented with. Starting with good quality ground truth images, the system attempts to produce enhanced images as good as the ground truth data it is trained with. However, if the thresholding algorithm applied yields an initial binary image that has large significant portions of the foreground wiped out as background, this will definitely negatively impact and limit the performance of the LUT classifier system regardless of having good quality ground truth images.

3.3 Learning phase: building the lookup table (LUT)

Suppose we have an image pair in our expert-labeled data set $T = \{(D, G)\}$, where D is the binary degraded document image, and G is the corresponding ground truth image. Let N represent an arbitrary $w \times w$ neighborhood bit pattern in D with p_i representing its center pixel located at position (x, y), while p_o denote the pixel at same position (x, y) in G. Let $p(x, y)$ represent the pixel value at (x, y), and $b(x, y)$ the binary code for the neighborhood N centered at (x, y). The binary code $b(x, y)$ is given by:

$$b(x, y) = \sum_{j=0}^{w^2-1} b_j * 2^j \tag{2}$$

where $b_j(x, y)$ denotes the j-th bit of $b(x, y)$ and is defined as:

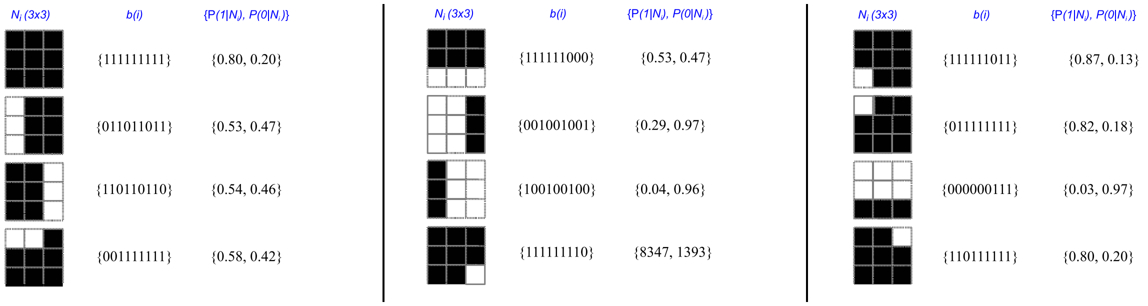
$$b_j(x, y) = p(x + L_x(j), y + L_y(j)) \tag{3}$$

where $L(j) \equiv (L_x(j), L_y(j))$ is the relative displacement of the j-th pixel in the neighborhood with respect to (x, y). The relative displacements are given by:

$$(L_x(i), L_y(i)) = ((i\%w - \lfloor w/2 \rfloor), (i/w - \lfloor w/2 \rfloor)) \tag{4}$$

where % denotes modulus division. e.g. for a 3 × 3 neighborhood: $L_x = [-1\,0\,1 - 1\,0\,1 - 1\,0\,1]$ and $L_y = [-1 - 1 - 1\,0\,0\,0\,1\,1\,1]$.

The binary code $b(x, y)$ (also denoted by $b(i)$) is the LUT key used to represent a neighborhood N_i with the center pixel p_i. Let $P(l|N_i)$ be the conditional probability of the output center pixel p_o at (x, y) in G being l (a foreground or background pixel i.e., $l \in [0, 1]$), given an input pixel p_i at (x, y) and its neighborhood N_i centered at (x, y) in D. The goal of the training phase is to obtain the data needed to estimate $P(l|N_i)$ for all neighborhood patterns found in D. For each input pixel p_i in D, we determine the binary code for its $w \times w$ neighborhood N_i and compute the values of $P(0|N_i)$ and $P(1|N_i)$. (We represent *foreground* pixels as 1 and *background* pixels as 0). We store $P(0|N_i)$ and $P(1|N_i)$ in a LUT indexed by $b(i)$, the binary code of N_i.



N_i (3x3)	$b(i)$	$\{P(1\|N_i), P(0\|N_i)\}$	N_i (3x3)	$b(i)$	$\{P(1\|N_i), P(0\|N_i)\}$	N_i (3x3)	$b(i)$	$\{P(1\|N_i), P(0\|N_i)\}$
	{111111111}	{0.80, 0.20}		{111111000}	{0.53, 0.47}		{111111011}	{0.87, 0.13}
	{011011011}	{0.53, 0.47}		{001001001}	{0.29, 0.97}		{011111111}	{0.82, 0.18}
	{110110110}	{0.54, 0.46}		{100100100}	{0.04, 0.96}		{000000111}	{0.03, 0.97}
	{001111111}	{0.58, 0.42}		{111111110}	{8347, 1393}		{110111111}	{0.80, 0.20}

Fig. 3 An example of the 12 most occurring entries in a lookup table (*LUT*) obtained using a 3 × 3 filter window. Each entry consists of unique neighborhood bit patterns N_i, denoted by its binary code $b(i)$, found in the degraded image and the matching probability set of $\{P(1|N_i), P(0|N_i)\}$.

The LUT is a mapping of all the unique patterns of N_i, using the key $b(i)$, existing in D to its probability set $\{P(1|N_i), P(0|N_i)\}$, as illustrated in Fig. 3. The neighborhood size $w \times w$ the LUT considers, can be viewed as the dimension of the filter window ($winsize$).

To build the LUT, we scan each pixel p in D to obtain its corresponding N except for two sets of pixels which we consider *non-relevant* for efficiency reasons. The first *non-relevant* set are all pixels for which we cannot obtain a complete neighborhood pattern in D, i.e., the boundary pixels located at positions

$$\{(x, y)|x\langle w/2 \vee x\rangle n - w/2 \vee y\langle w/2 \vee y\rangle m - w/2\} \quad (5)$$

where (m, n) are the image dimensions of D. This does not diminish the effectiveness of the classifier, as there are generally no foreground data contained in the border region of document images. The second set are all the pixels having a neighborhood of only white pixels, i.e., pixels at positions $\{(x, y)|b(x, y) = 0\}$. These pixels are not considered since N has no foreground data. This greatly reduces the number of pixels we must process in D. The algorithm to build a $LUT(w, T)$, given $T = \{(D, G)_j, j = 1, \ldots, t\}$, is summarized in Algorithm 1.

Algorithm 1 Build-LUT

```
Build-$LUT$(w, $T = \{(D, G)_j, j = 1, \ldots, t\}$)
$P(1|N_i) = 0$; $P(0|N_i) = 0$
**for all** $(D, G)_j \in T$ **do**
    **for all** *relevant* $p_i(x, y) \in D$ **do**
        obtain $N = b(x, y)$
        **if** $p_o(x, y) == 1$ **then**
            $P(1|N_i) + 1$
        **else** $\{p_o(x, y) == 0\}$
            $P(0|N_i) + 1$
        **end if**
    **end for**
**end for**
$P(p|N_i)$ normalized such that $0 \le P(p|N_i) \le 1$
end buildLUT
```

3.4 Correction phase: LUT classification

During the correction phase, we apply the LUT classifier to a given degraded document image $D \notin T$ (i.e., the expert-labeled dataset) to obtain its enhanced image $\hat{G}$. The basic LUT classifier is an ensemble of two classifiers: (i) Approximate Nearest Neighbor (ANN) classifier, and (ii) Maximum Likelihood (M-L) decision classifier. To enhance D given a $LUT(w, T)$, we scan each pixel $p_i \in D$, using the filter window size w, to compute a binary key $b(i)$ (The same set of pixels ignored during the learning phase are also ignored during the correction phase). Using the key $b(i)$ and the LUT, we obtain its corresponding probability set $\{P(0|N_i), P(1|N_i)\}$ to determine p_o. The core of our algorithm lies on the premise that we can estimate the probability of output of a single pixel p_i in its enhanced image by taking into account the spatial information obtained from its surrounding pixel neighborhood centered on p_i. There are two main steps in the correction process: the first is the *lookup operation* of the LUT entry N defined by the binary key, handled by the ANN classifier, and secondly, the pixel classification decision of the output center pixel, performed by the M-L classifier. Both steps are described in detail later.

3.4.1 Approximate nearest neighbor cluster classifier.

During the correction process, it is important that our LUT can generalize well to be able to process unseen samples (i.e., values of N not encountered during the training phase). For example, if we have a 5 × 5 neighborhood, even a small difference in one pixel out of the 25 total pixels can cause the lookup operation of N in the LUT to fail, if the slight variation was not trained for. To overcome this, we perform the lookup operation using an ANN classifier which utilizes the k-Nearest Neighbors Search Algorithm by Arya et al. [7] to search for similar entries to the unseen sample. ANN performs approximate nearest neighbor searching, based on the use of standard and priority search in kd-trees and balanced box-decomposition (bbd) trees. The ANN classifier returns the probability set of N, if N is found in the LUT, or the probability sets for k most similar entries of N found in the LUT. This output is passed on to the M-L Classifier to make a pixel classification decision. Thus, the lookup operation classifies each pattern of degradation i.e., N to exactly the same or k most similar patterns of N existing in the LUT.

Each entry N in a given LUT, represented by its binary code, given by Equation (2), are preprocessed by ANN into a kd-tree [18] data structure. To compute the similarity distance for any two entries, ANN uses the Euclidean distance between their binary codes. For any query point $N \notin$ LUT, the ANN classifier is able to report the k nearest entries with ϵ approximation to N efficiently. The ϵ specifies the maximum approximation error bound, which permits us to control the trade-off between accuracy and running time. We show the impact of both ANN parameters, k and ϵ, on the running time and accuracy of the LUT classifier in Sect. 4.2.1.

3.4.2 *Maximum likelihood classifier*

The M-L classifier makes a pixel classification decision based on the conditional probability of the output pixel $P(p_o|N_i)$, as defined in Sect. 3.3, using the probability set information of N obtained from the ANN classifier.

$$p_o(x, y) = \underset{p\in\{0,1\}}{\operatorname{argmax}} P(p|N) \tag{6}$$

The computation of the value of the output center pixel given its neighborhood information N is essentially the maximum likelihood estimate of $p_o(x, y)$ being a foreground or a background pixel using the conditional probability information $\{P(1|N_i), P(0|N_i)\}$. Given the probability of $p_o(x, y) \in G$ being 1 or 0 for N during training, we estimate the value of $p_o(x, y) \in \hat{G}$ to be 1 if $P(1|N_i) > P(0|N_i)$, and vice-versa for 0. If $P(1|N_i) = P(0|N_i)$, we take no action: $p_o(x, y) = p(x, y)$.

The ANN classifier may determine k neighbors. It sends the probability set information for a set $\{N_r, r = 1, \ldots, k\} \subset LUT(w, T)$ to the M-L classifier. If $k > 1$, the pixel classification decision of $p_o(x, y)$ is based on the majority vote over the set $\{N_r, r = 1, \ldots, k\}$. For each N in the set, we obtain its estimate of $p_o(x, y)$ using Equation (6) and then compute the majority vote over the individual estimates obtained. If there is no majority, no action is taken. The correction process is summarized in Algorithm 2.

3.5 Performance of LUT classifier

Theoretically, the size of the LUT ($\|\{N\}\|$) is bounded by $\boldsymbol{O}(2^{w^2})$ as $N_i \equiv b(i)$ has a length of w^2. This implies an exponential memory requirement which will translate to a very inefficient system. For example, using a $w - 5$ filter for an LUT would require a memory storage of about 33 MB (2^{25}) while for $w - 7$ filter 524288 GB (2^{49})! Intuitively, the actual bound of the LUT will be much less given that not all possible pixel pattern configurations will exist in typewritten document images.

Algorithm 2 Correct-D to obtain $\hat{G}$

```
Correct-$D$(w, LUT, $\epsilon$, $k$)
arrange LUT into ANN structure with parameters $\epsilon$ and $k$
for all *relevant* $p_i(x, y) \in D$ do
  obtain $N = b(x, y)$
  if $N \in$ LUT then
    ANN Classifier returns $\{P(1|N), P(0|N)\}$ for $N$
    set $p_o(x, y) \in \hat{G} = 0/1/p_i(x, y)$ using equation 6
  else $\{N \notin \text{LUT}\}$
    $vote\,0 = 0$; $vote\,1 = 0$ //counters for majority voting
    ANN Classifier returns $\{P(1|N_r), P(0|N_r)\}$ for $\{N_r, r = 1, \ldots, k\}$
    for $r = 1$ to $k$ do
      // (using equation based on $N_r$)
      if $p_o(x, y) == 0$ then
        $vote\,0 + 1$
      else if $p_o(x, y) == 1$ then
        $vote\,1 + 1$
      end if
    end for
    //take majority vote to set $p_o(x, y) \in \hat{G}$
    if $vote\,0 > vote\,1$ then
      $p_o(x, y) = 0$
    else if $vote\,1 > vote\,0$ then
      $p_o(x, y) = 0$
    else $\{vote\,1 == vote\,0\}$
      $p_o(x, y) = p_i(x, y)$
    end if
  end if
end for
end Correct-$D$: output $\hat{G}$
```

To validate this assumption we measured the number of different neighborhoods occurring in actual documents images. We used a set of 25 document image pairs to observe the size of the LUT for $w = 5, 7, 9$. From the experimental results, we observed that a small percentage of all the possible bit patterns exist in document images. The percentage of entries to the total number of theoretically possible entries actually decreased exponentially as w increased. Therefore, the empirical bound on the size of the LUT is $\ll 2^{w^2}$. Our experiments, as discussed in Sect. 4.2.3, demonstrate that a small set of images is sufficient to learn the degradation and enhancement patterns. To improve the performance of the LUT classifier, we utilize the map container data structure. The performance of lookup operation for each N is $\boldsymbol{O}(\log(||T||))$.

3.6 Cascade LUT classification

We can further improve the performance of the LUT classifiers by applying the classifiers in a cascaded configuration. When training a basic LUT classifier, as described in

Sect. 3.3, we compare a degraded binary document image D to its ground truth image G, given $T = \{(D, G)_i, i = 1, \ldots, t\}$, to produce a single LUT. In the cascade LUT classifier configuration, we produce multiple LUTs during the learning phase using the same expert-labeled data set T.

Let LUT_1 denote the first LUT obtained by comparing each $D \in T$ to its corresponding G. We apply LUT_1 on each $D \in T$ to obtain its estimated corrected image $\hat{G}$. We then build LUT_2 using $T' = \{(\hat{G}, G)_i, i = 1, \ldots, t\}$. We compare the output image $\hat{G}$ resulting from applying LUT_1 on the degraded binary image D to the ground truth image G to obtain LUT_2. A two-stage cascade LUT classifier comprises of LUT_1 and LUT_2. To correct a document image $D \notin T$, we apply LUT_1 and LUT_2 in the sequential order they were built. Thus, we apply LUT_1 initially to D to get $\hat{G}_1$, then we apply LUT_2 on $\hat{G}_1$ to obtain $\hat{G}$, which is the final corrected image of D given by the cascade configuration.

The goal of the cascade is that, with each stage, the next LUT improves on the work done by the previous LUT. Each stage in the cascade attempts to correct the more difficult points to classify in the original document. There is an additional overhead cost of increased execution time—twice the cost of using a single LUT.

The cascade LUT classifier can be generalized to comprise m LUTs representing a set of m classifiers applied in sequential order. When building a m-cascade LUT classifier, the process is terminated if during the iterations of training new LUTs, we obtain an LUT_{i+1} that yields no improvement on the training data compared to the former LUT_i. The performance of the cascaded LUT classifier is discussed in Sect. 4.3.

4 Experimental results and analysis

4.1 Experimental setup

To evaluate the effectiveness of the proposed approach, we evaluated performance on a subset of document images drawn from the Frieder diaries collection [17]. We obtained the ground truth images for our test dataset by employing human experts to label each image using an interactive document enhancement software [10]. The ground truth generation process is outlined in Sect. 3.1. Each document image is approximately 1,200 by 1,750 pixels in size and contains 2,600 character instances on average. Thus, a total of 14 document images is equivalent to approximately 36,400 characters. It should be noted that the images used in the training dataset is separate from the test data set. We performed character segmentation on each document image prior to applying the filter to ensure that as we scan the document image pixel by pixel, the filter window does not overlap neighboring characters. The filter ignores any neighboring character's pixel information contained in its window.

The experimental results demonstrated in this section have three main purposes: (1) to validate the effectiveness and efficiency of the LUT classifier's ability to learn and correct patterns of degradation; (2) to compare the LUT classifier to other known classifiers such as the SVM; (3) to compare the enhancement performance of the LUT classifier to other state of the art enhancement techniques using OCR accuracy measure.

The efficiency of a classifier was measured by its execution time in seconds. We performed all the experiments on a standard PC machine with Intel Core(TM)2 CPU 1.67 GHz processor and 2.0 GB memory.

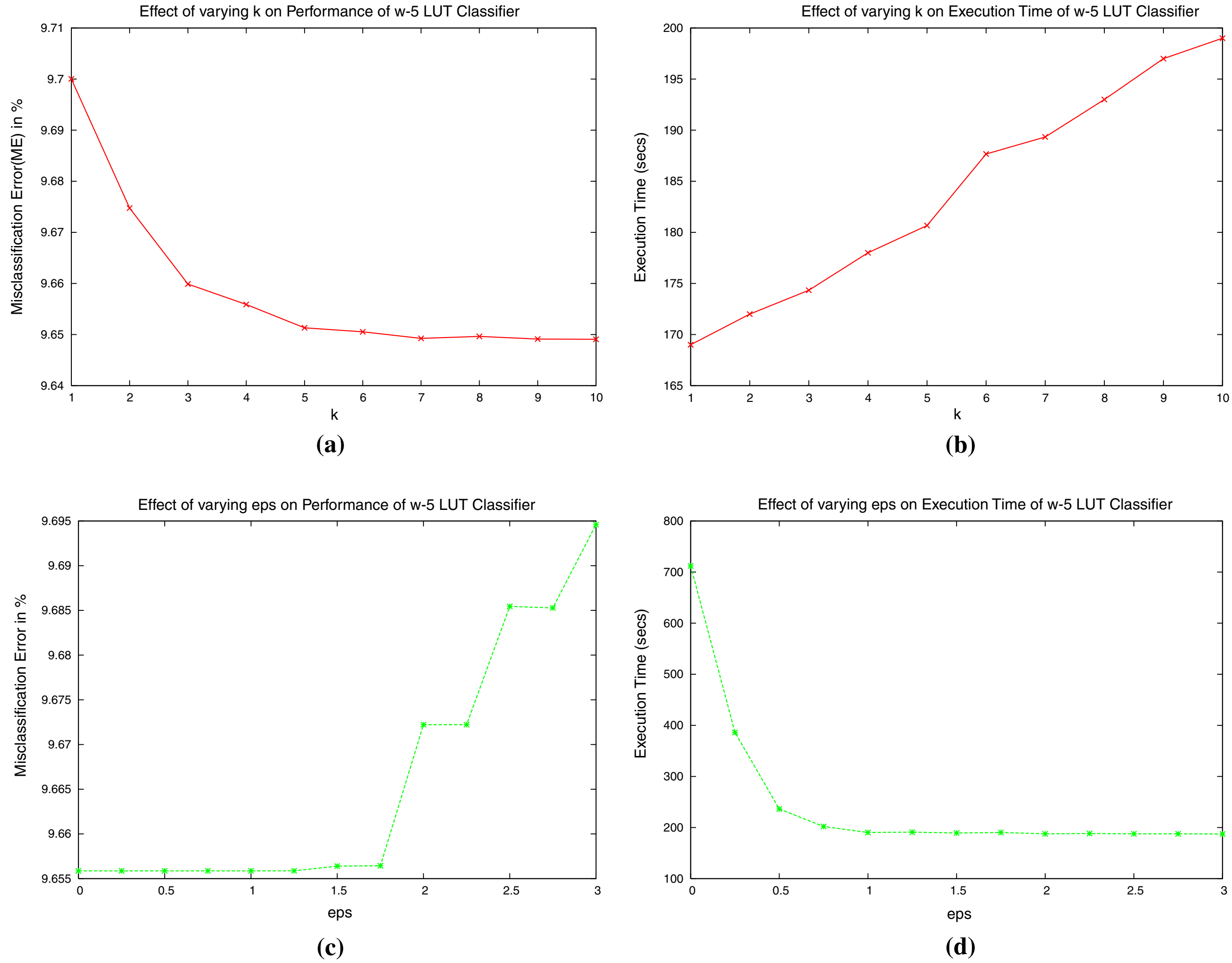
Given that our goal is to enhance the degraded document image and the ground truth image is the perfect standard of enhancement, we use the Misclassification Error (ME) [27] as the performance criteria for validate the effectiveness of the LUT classifier. ME is defined as $(M/P) \times 100$, where M is the number of pixels in the output image $\hat{G}$ that do not correlate with the ground truth image G and P is the number of pixels in the original binary degraded image D. We also perform a qualitative analysis of the results obtained by observing them visually to validate that there is actually an improvement in the enhancement. We define the *base*-ME as the value of ME obtained by comparing the binary image (obtained after preprocessing) to its ground truth image before we apply the classifier to the image. The *base*-ME, which is computed using segmentation without correction, enable us to quantify how much improvement is obtained by the LUT classifiers beyond basic foreground–background separation.

4.2 Systemic evaluation of the LUT classifier's parameters

The LUT classifier has four parameters: the ANN parameters: k (number of nearest neighbors) and ϵ (approximation error), the dimensionality of its filter window $winsize$ (w), and the size of the training dataset used to build the lookup table. There is an accuracy-time trade-off cost associated with selecting the optimal values for the parameters.

4.2.1 ANN parameters

The goal of the initial set of experiments was to determine the optimal values of k and ϵ that give us the best performance in terms of accuracy at a reasonable cost in execution time. As we observe from the results, the ANN parameters do not impact the performance of the classifier significantly compared to their effect on the execution time. Figure 4 shows the performance of the LUT classifier as a function of ANN parameters with a fixed $winsize$ of 5. From Fig. 4b, we observe that for a fixed value of ϵ, as we increase the number of neighbors k, we obtain a lower ME at an increased cost



(a) **(b)**

(c) **(d)**

Fig. 4 Selecting the optimal ANN parameters k and ϵ: Trade-off between performance and execution time. **a** Performance (vs.) k, ϵ fixed at 1.25, **b** execution time (vs.) k, **c** performance (vs.) ϵ, k fixed at 4, **d** execution time (vs.) ϵ.

of execution time. We also observe that as the k increases beyond 4, the marginal decrease in ME is small while the execution time still steadily increases. Therefore, using both time and performance constraints, we chose the optimal value k as 4 for our experiments. Similarly from Fig. 4c and d, we observe that the distance approximation error bound parameter ϵ does not impact the performance (as given by ME) of the classifier significantly compared to its effect on execution time. Using both graphs, we selected the optimal value of ϵ as 1.25. Thus, for all subsequent experiments, we fixed k and ϵ at 4 and 1.25, respectively.

4.2.2 *Dimensionality of filter window $w \times w$*

The next set of experiments evaluate the performance and efficiency of the LUT classifier with respect to its *winsize* parameter. Figure 5 shows the performance of the LUT classifier for three different $winsize$ values {5, 7, 9} using the same training dataset. As can be observed from Fig. 5 (by comparing the ME obtained by the LUT classifiers to the *base*-ME), we reduce the error greatly beyond Bernsen segmentation. Using a w-9 filter, we obtain an average ME of 7.690% which is a 51.6% improvement over the average *base*-ME of 15.893%. This demonstrates the effectiveness of the proposed approach compared to the initial segmentation technique (Bernsen's method [11]). By using the LUT classification method, we advance the initial binarization technique by 51.6%. This implies that we could similarly improve on other state-of-the-art segmentation techniques by plugging them in as our initial binarization technique.

From Fig. 5, we observe that the w-9 classifier attains the best performance on enhancement of degraded images. We can observe that as we increase the dimensionality of the filter window, the enhancement performance increases as quantified by the decrease in ME. The w-5 filter yields an average ME of 9.656% compared to 7.690% obtained by the w-9

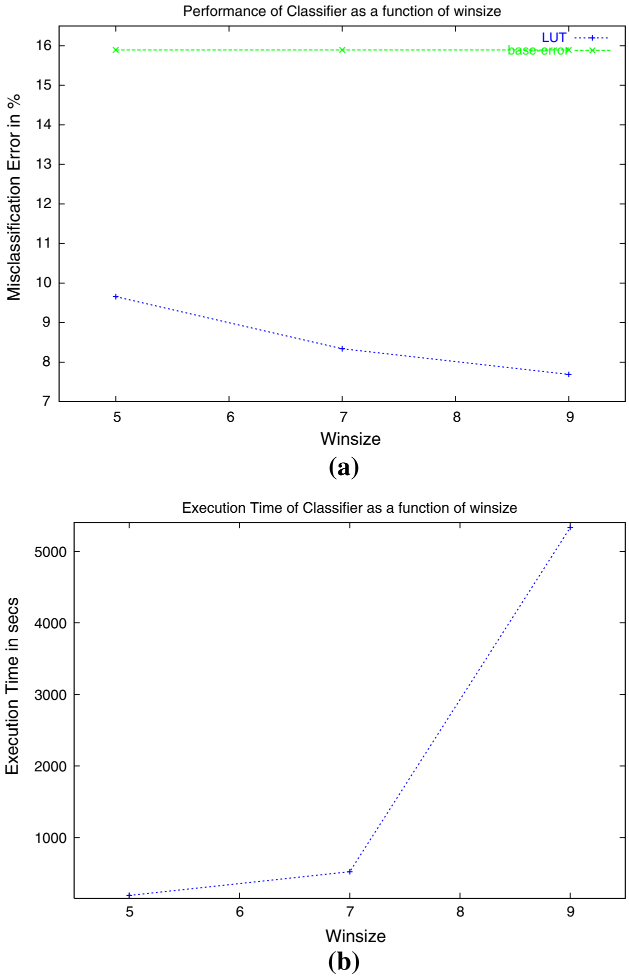


Fig. 5 Effect of winsize value (5, 7, 9) on effectiveness and complexity of LUT classifiers. **a** Performance of LUT (vs.) winsize, **b** execution time as a function of winsize

filter. However, an increase in *winsize* also results in a greater computational cost that affects the execution time of the classifier, as shown in Fig. 5. The average execution time per set of 10 document imagesusing a w-9 filter is 5332s, while for a w-5 filter, it is 191s.

Qualitative results of one of the test images are shown in Fig. 6. In the enhanced output obtained by the w-5 filter, as shown in Fig. 6, we can observe that character such as ‘e’s and ‘s’s that exhibit filled character degradation are much clearer compared to the initial segmented document image in Fig. 6. However, some of the ‘e’s and ‘s’s of the w-5 filter enhanced output are still closed. The majority of these errors are removed in the output images of both w-7 and w-9 filters. We can observe that in the output image of the w-9 filter (as shown in Fig. 6), the majority of the noise in characters have been filtered out. The characters in the output image of the w-9 filter are much clearer and distinct though some are still slightly broken. The output image obtained by the w-9 filter visually matches the ground truth image better compared to the other two filters. The visual results validate the ME performance criteria measure as we observe a correlation between improved enhancement and a low ME value.

The results presented show that the proposed LUT classifiers are an effective way of learning patterns of degradations and their correction directly from the data. The strength of our approach also lies in its generality; the method is general given that the degradation model is based on actual degraded images.

4.2.3 Size of training dataset

The next set of experiments evaluate the effect of increasing the size of the training dataset on the performance of the LUT classifiers (sample complexity). From Fig. 7, we can observe, as expected intuitively, that as we increase the training set size T, the performance of the classifier improves though the marginal improvement decreases. The capacity of the w-5LUT reaches a saturation point more rapidly than that of the w-7LUT. This is because a LUT with a larger $winsize$ parameter has greater capacity and so can take more information before reaching a saturation point. We can also observe that training a LUT classifier based on a few expert images is sufficient for the classifier to learn patterns of degradation and correction.

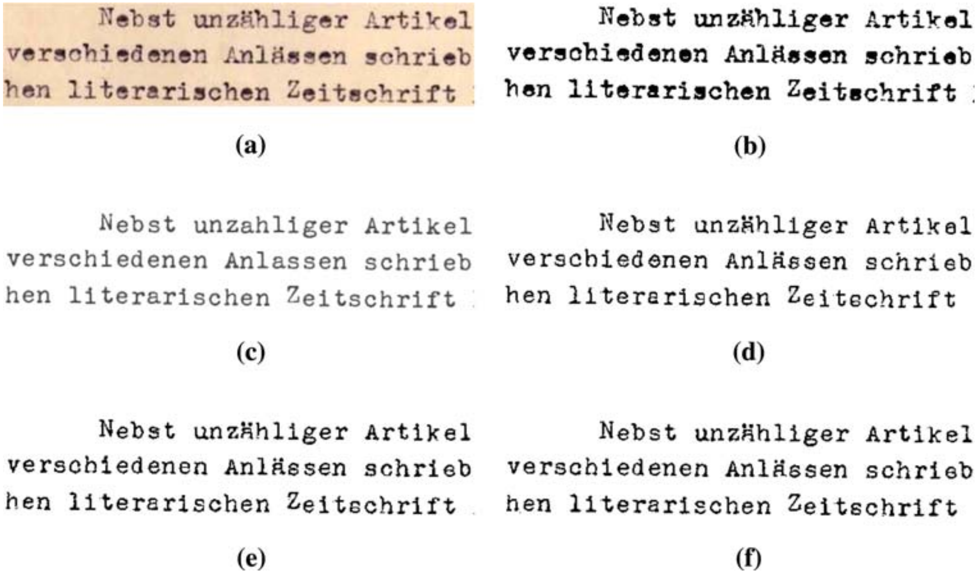
The visual results of applying the w-7LUT with two different dataset sizes $T = 1, 7$ is shown in Fig. 8. We observe from Fig. 8, that the characters in the enhanced output image obtained with a w-7LUT trained with the larger dataset ($T = 7$) are more defined compared with the enhanced output image using a smaller dataset ($T = 1$). For example, the ’e’s in Fig. 8c are not closed up as in Fig. 8b. We conclude that the higher the capacity of the LUT used, the greater its ability to learn from a larger number of expert images without suffering from over-fitting of training data.

4.3 Performance of cascade LUT classifiers

To evaluate the performance of arranging the LUT classifiers in a m-cascaded configuration, we varied the number of stages from 1 (i.e., no cascade) to 10 for both w-5 and w-7 filters. The performance of the m-stage cascaded LUT classifiers improves compared to using a single stage classifier, as shown in Fig. 9. For a w-5LUT, the two-stage cascade results in a 2.5% improvement over a basic configuration i.e., with no cascade. We also observe from Fig. 9, as expected, that the increase in performance going from no cascade to a two-stage cascade is more pronounced for a w-7LUT.

There is a bound on the number of stages m that leads to improved performance, as can be observed from the graph. This is because during the learning phase, the result obtained after a single stage is already close to the ground truth image. The classifier is able to learn the training data images almost

Fig. 6 Result of applying LUT classifier of different *winsizes* on a test image. **a** Degraded document image, **b** segmented degraded image, **c** ground truth, **d** w-5filter enhanced output, **e** w-7filter enhanced output, **f** w-9filter enhanced output



Nebst unzähliger Artikel
verschiedenen Anlässen schrieb
hen literarischen Zeitschrift

(a)

Nebst unzähliger Artikel
verschiedenen Anlässen schrieb
hen literarischen Zeitschrift

(b)

Nebst unzahliger Artikel
verschiedenen Anlassen schrieb
hen literarischen Zeitschrift

(c)

Nebst unzähliger Artikel
verschiedenen Anlässen schrieb
hen literarischen Zeitschrift

(d)

Nebst unzähliger Artikel
verschiedenen Anlässen schrieb
hen literarischen Zeitschrift

(e)

Nebst unzähliger Artikel
verschiedenen Anlässen schrieb
hen literarischen Zeitschrift

(f)

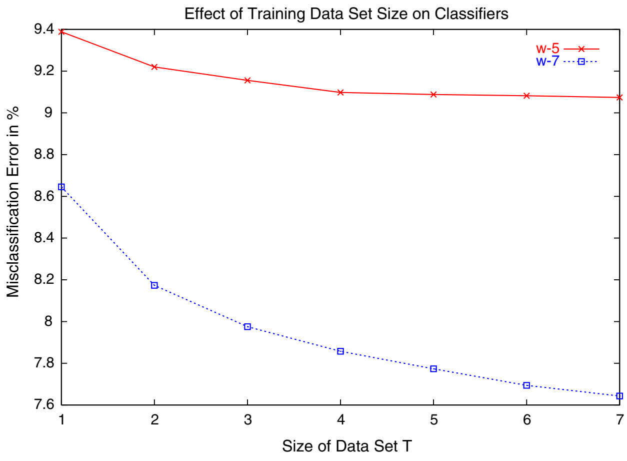
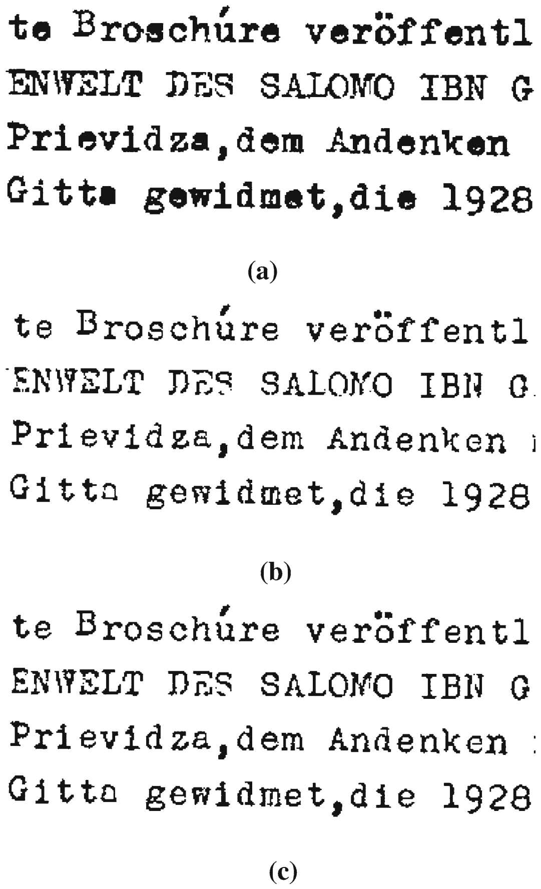


Fig. 7 Size of training data set (vs.) performance of the LUT classifier

perfectly especially for the LUTs with larger *winsize* values such as a w-7filter. The additional LUTs thus provides little or no information in correction of the degradation. As can be observed from Fig. 9, beyond three stages of a w-5LUT classifier, there is not much improvement of classification error. This is due to the large capacity of the individual stages. For the w-7LUT classifier, the best performance is attained with two stages and is better than that of the w-5LUT classifier. This is because the capacity of the w-7LUT is large and so does not benefit much from subsequent cascades. The marginal performance drop with more than two cascade stages with the w-7LUT classifier may be due to over-fitting of the training data.



te Broschúre veröffentl
ENWELT DES SALOMO IBN G
Prievidza,dem Andenken
Gitta gewidmet,die 1928

(a)

te Broschúre veröffentl
ENWELT DES SALOMO IBN G
Prievidza,dem Andenken
Gitta gewidmet,die 1928

(b)

te Broschúre veröffentl
ENWELT DES SALOMO IBN G
Prievidza,dem Andenken
Gitta gewidmet,die 1928

(c)

Fig. 8 Result of applying w-7LUT classifier with different training set sizes ($T = 1, 7$) on a test image. **a** Segmented degraded image, **b** enhanced output, $T = 1$, **c** enhanced output, $T = 7$

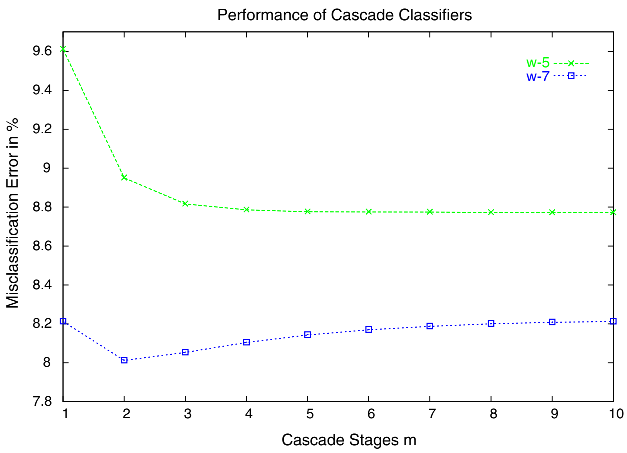


Fig. 9 Performance of cascaded LUT

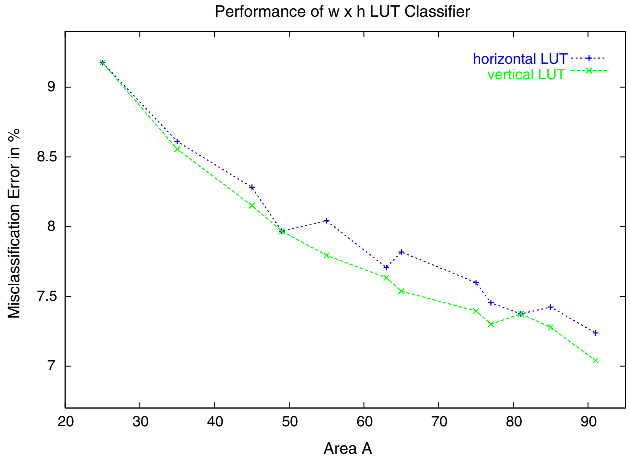


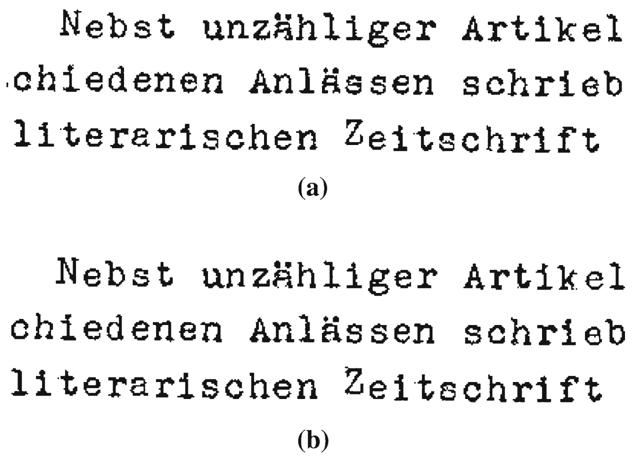
Fig. 10 Performance of $w \times h$ LUT

4.4 Generalization to $w \times h$ filters

Thus far, we have evaluated the performance of LUT classifiers using square filter windows w. From the experimental results obtained, we observe not only that a w-9filter (with the largest pixel neighborhood area) yielded the best performance, but also had the longest execution time. In this section, we evaluate the correlation between the effectiveness of the LUT classifier and the size of the pixel neighborhood area A, by generalizing $w \times w$ to non-square filter windows $w \times h$, where w denotes the width and h the height.

It is possible when using non-square windows for two filters to have the same area A but reversed window configuration. Hence, given two filters a and b with the same area A such that a considers a neighborhood $w \times h$ while b has $h \times w$, if $w > h$, filter a is a *horizontal* filter and b is a *vertical* filter. In the following evaluation, we varied both values of w and h, where $w \in \{5, 7, 9\}$ and $h \in \{5, 7, 9, 11, 13, 17\}$. We also distinguish between vertical filters and horizontal ones.

From the results obtained, as shown in Fig. 10, vertical filters produce a lower ME compared to their horizontal counterparts. This implies that viewing a longer section of the pixel neighborhood than a wider section improves pixel classification possibly because it provides a more discriminant information. For most characters, the height is greater than the width. Hence, increasing the filter view height wise would usually produce more information than increasing it width wise. We also observe that, generally as the area increases, the performance of the LUT improves. However, the computational cost increases as well. A visual result of the enhanced output obtained from applying a 7 × 13 filter and its horizontal counterpart is shown in Fig. 11. Thus, we can conclude that the larger the pixel neighborhood considered by the LUT classifier, the better its enhancement performance at the cost of increased complexity as evident by the increased execution time.



Nebst unzähliger Artikel
chiedenen Anlässen schrieb
literarischen Zeitschrift

(a)

Nebst unzähliger Artikel
chiedenen Anlässen schrieb
literarischen Zeitschrift

(b)

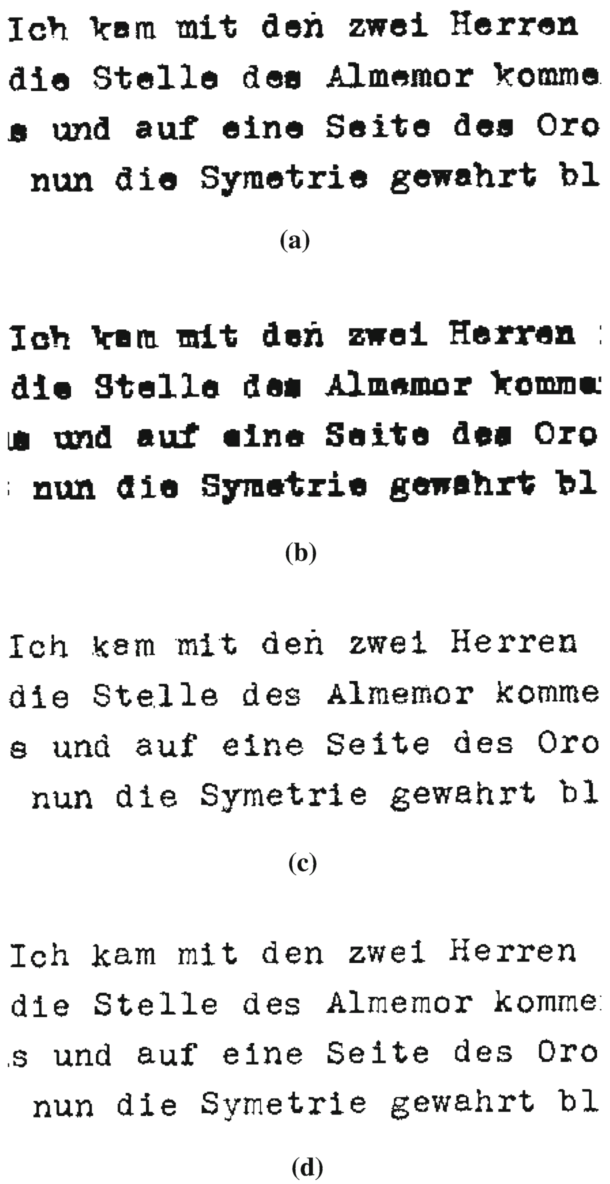
Fig. 11 Result of applying $w \times h$ horizontal and vertical LUTs on test image in Fig. 6a. **a** 7 × 13filter enhanced output, **b** 13 × 7filter enhanced output

4.5 Comparison to Zheng's LUT-based restoration algorithm

As mentioned in Sect. 2, Zheng et al. in [32] design a lookup table(LUT) for restoring a class of degraded document images. These are uniform text documents corrupted by various types of noises during the document generation and copying processes. Their method is based on the assumption that the degradation in the degraded images can be estimated by a set of parameters using the Kanungo morphological degradation model [21]. The LUT used is a 512 × 512 matrix built during training using a 3 × 3 filter window. For each 3 × 3 neighborhood pattern in the degraded image, all possible occurrences of the corresponding ideal output pattern in the ideal image and their probabilities are stored. During restoration, each patch in the degraded image is replaced with the most occurring output pattern encountered during training.

Both approaches are similar in that they are model-based restoration algorithms. However, our proposed LUT algorithm advances on the LUT image restoration work done by Zheng et. al in three main ways:

1. *The class of degraded document images*: Our focus is historical typewritten documents that consist of more pronounced degradation patterns than those that exist in scanned or copied uniform text document images. Thus, our algorithm attempts to learn these degradation patterns and their correction for each class of similar degraded images rather than estimate them using the Kanungo degradation model. The Kanungo degradation model is suitable for small perturbations [9] encountered during photocopying and scanning of uniform text documents but not to large degradations found in old typewritten documents.
2. *Design of LUT*: Our proposed algorithm builds a LUT that stores the probability of the center output pixel in the ideal image being a foreground or a background given a $w \times w$ pixel neighborhood in the degraded image. We correct one pixel at a time, taking into account the neighborhood information. In contrast, the method by Zheng et. al stores the probability of the 512 possible pattern occurrences of output patch in the ideal image given a 3×3 pixel neighborhood in the degraded image. The core of our algorithm lies on the premise that we can best estimate the probability of output of a single pixel p in its enhanced image by taking into account the spatial information obtained from its surrounding pixel neighborhood centered on p. We also employ the use of the k-Nearest Neighbors Search Algorithm during the classification process to ensure that our lookup table generalizes well in classifying samples that are not found in the table.
3. *Degradation model/dataset*: Actual degraded images are employed in the design and evaluation of our proposed LUT algorithm. We generate the degradation model using these real degraded images, while Zheng's restoration algorithm uses images synthetically degraded to estimate the parameters of the Kanungo degradation model.



Ich kem mit deň zwei Herren
die Stelle des Almemor komme
s und auf eine Seite des Oro
nun die Symetrie gewahrt bl

(a)

Ich kem mit deň zwei Herren
die Stelle des Almemor komme
s und auf eine Seite des Oro
nun die Symetrie gewahrt bl

(b)

Ich kem mit deň zwei Herren
die Stelle des Almemor komme
s und auf eine Seite des Oro
nun die Symetrie gewahrt bl

(c)

Ich kam mit den zwei Herren
die Stelle des Almemor komme
.s und auf eine Seite des Oro
nun die Symetrie gewahrt bl

(d)

Fig. 12 Application of Zheng's et al. LUT design on historical degraded document images in comparison with proposed LUT method. **a** Degraded binary image, **b** 3×3 Zheng's LUT enhanced output; Mean ME= 22.31%, **c** proposed w-9LUT output image; Mean ME = 7.69%, **d** ground truth image

To evaluate the effectiveness of the Zheng's LUT design on our class of degraded documents, we trained a LUT using a 3×3 filter to obtain a 512×512 matrix based on the same expert-labeled dataset used for our experiments in Sect. 4. We applied the LUT to restore the degraded images in our historical typewritten documents dataset by replacing each patch in the degraded image with its most probable ideal patch from the LUT. As can be observed from the visual results displayed in Fig. 12, the enhanced image generated by the LUT actually appears a little more degraded with strokes and lines filled in where they not ought to be. The LUT, however, appears to be effective at smoothing the pixel noise around some characters.

To evaluate the performance of the Zheng's LUT quantitatively in comparison with the w-9LUT, we utilized the mean ME measure. As noted in Fig. 12, the mean ME obtained for the Zheng's LUT is 22.3% compared to 7.69%, the mean obtained for the w-9LUT. Thus, using quantitative measures, the proposed LUT far outperforms the Zheng's LUT. This implies that our proposed LUT is more capable to learn patterns of correction of degradation to yield an enhanced image that is similar to the ground truth image.

A key component missing here is the estimated parameters for the morphological degradation model that their method uses. This may explain why Zheng's method is not effective in this experiment. Automatically, estimating the parameters of the morphological degradation model on a large and varied degradations as in our class of degraded images is a non-trivial task.

4.6 LUT classification in comparison with known classifiers

A supervised classification task involves training and testing datasets and a learning algorithm. In the proposed LUT algorithm, we handle the process of learning and correcting degradations historical documents as a three-class classification problem. We compare the performance of our LUT classifier to that of a SVM, which is one of the best known general purpose learning algorithms. The purpose of the following experiment is to validate the proposed LUT algorithm as a simple yet efficient classification design that is tuned to large descriptors by comparing it to a known classifier, SVM. The goal of SVM is to produce a model that predicts target values of data instances in the testing set given feature attributes [12].

We applied the SVM classification algorithm to our degradation model using LIBSVM, a library for support vector machines [14]. We set the $winsize$ parameter as 5, our simplest filter configuration. Using the same training dataset, we applied both the SVM and LUT algorithms to generate their individual models. We then applied the model to the test dataset and evaluated its pixel accuracy, compared to the ground truth images, as quantified by the ME measure. From the results obtained, as shown in Table 1 and Fig. 13, we can observe that the LUT classifier yields better performance. In contrast to known classifiers such as the SVM that are complex, the LUT classifier is simple and able to deal with the inherent ambiguity in the training dataset in the application of document enhancement.

Table 1 Summary of LUT (vs.) SVM classifier performance

	SVM	LUT
Training time (min)	58.7	5
Execution time (min)	16.3	1.85
Misclassification error (%)	10.71	9.66

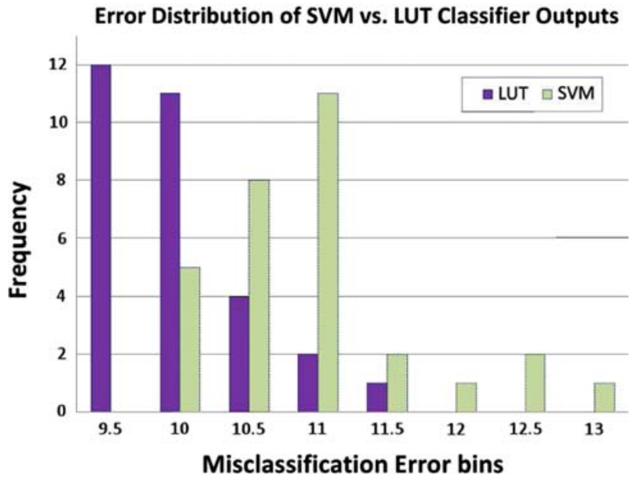


Fig. 13 Histogram of the misclassification error distribution of the SVM classifier (vs.) LUT classifier

4.7 Comparison to binarization and postbinarization techniques

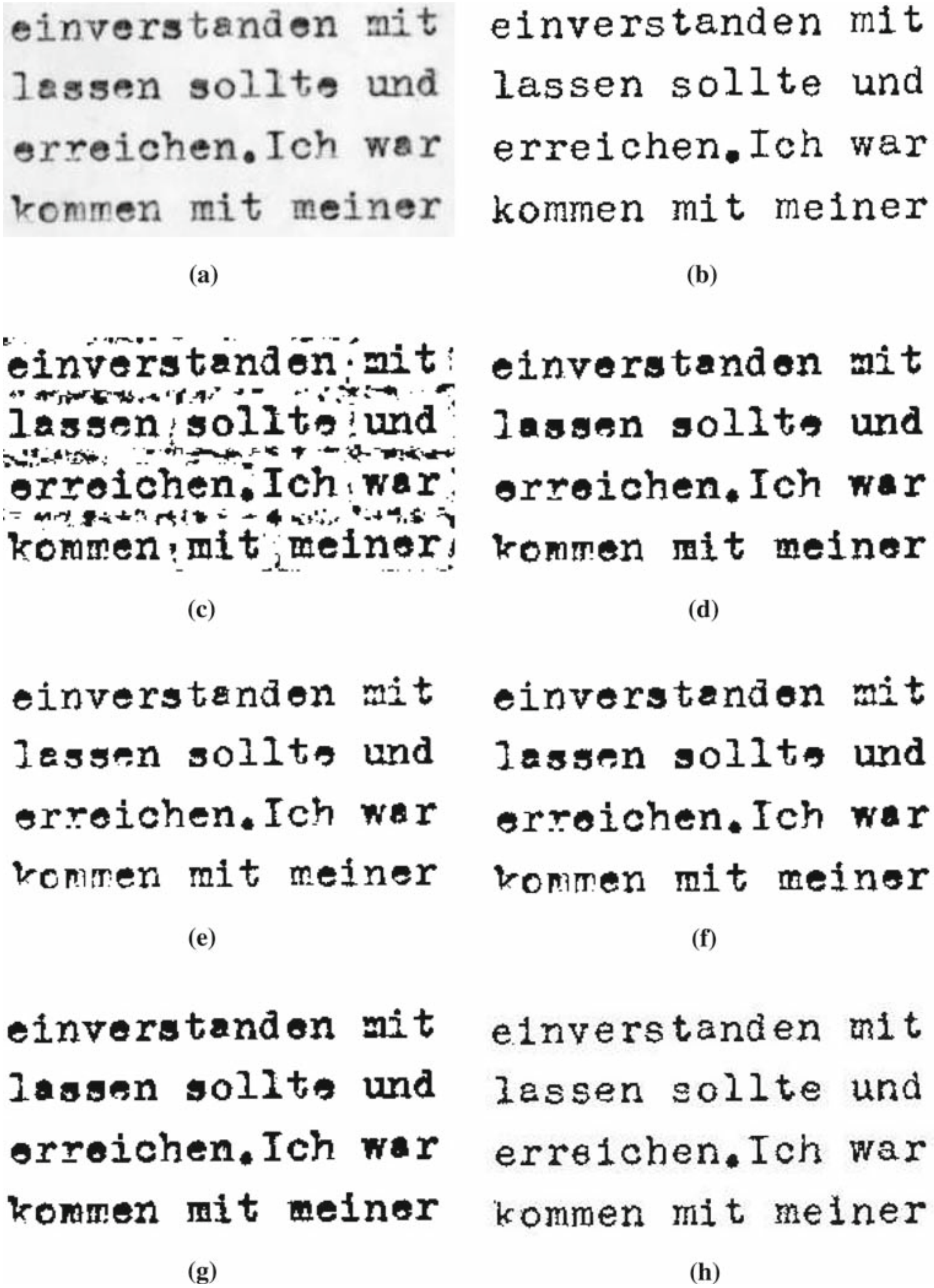
4.7.1 Binarization techniques

We compared the performance of our algorithm with four well-known binarization techniques in addition to the Bernsen method [11] applied in our algorithm. We evaluated the following: Otsu's global thresholding method [24], Niblack's adaptive thresholding method [23], Sauvola et al. adaptive method [26], and Gatos et al. adaptive degraded document method [19]. We implemented the Gatos method using the plugin provided by the Gamera's software [15].

As can be observed from the visual results shown in Fig. 14, the LUT classifier algorithm outperforms the other algorithms in its ability to enhance the degraded image by correcting the large degradations in the image. The readability of the images is improved by employing actual degradation models in the enhancement process.

To perform a quantitative analysis of the different approaches, we utilized the ME measure in comparison with the ground truth image. Table 2 lists the mean ME obtained for each method. (The Niblack method outperforms the Otsu only because the ME measure focused on the labeled foreground areas in the document image. However, visually we see that the Niblack method creates a lot of noise in the predominantly background areas which other methods do a good job of minimizing.)

It is also interesting to observe the performance of our algorithm in comparison with the other techniques in terms of OCR accuracy. We used the Tesseract (2.04) OCR engine [28] to convert all the binary images obtained from each algorithm to text output. Given that typewritten documents


einverstanden mit
lassen sollte und
erreichen. Ich war
kommen mit meiner

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

Fig. 14 Result of applying the different algorithms on a test degraded image. **a** Original degraded image, **b** ground truth image, **c** Niblack, **d** Otsu, **e** Sauvola et al., **f** Gatos et al., **g** Bernsen **h** w-9LUT

are not usually aligned properly, we were only able to apply the engine to a subset of the test dataset. We generated the ground truth text for each document image using the interactive document enhancement software. There was a total of 14,203 characters in the set of document images tested. It should be noted that the test document images have a relatively low resolution of 72 ppi.

We quantified the OCR accuracy using the Levenshtein Distance (LD) measure, similar to the evaluation method employed by Gatos et al. in [20]. As we can observe from the chart in Fig. 15, the ground truth images had the lowest LD measure, followed by the LUT algorithm , the Niblack method and Gatos. The lowest performance was obtained by Otsu—a global thresholding method. The Niblack method

Table 2 Quantitative analysis of the binarization techniques using misclassification error (ME)

Method	Mean ME in %
Otsu	16.56
Niblack	15.54
Sauvola et al.	11.58
Gatos et al.	12.5
w-9 LUT	7.69

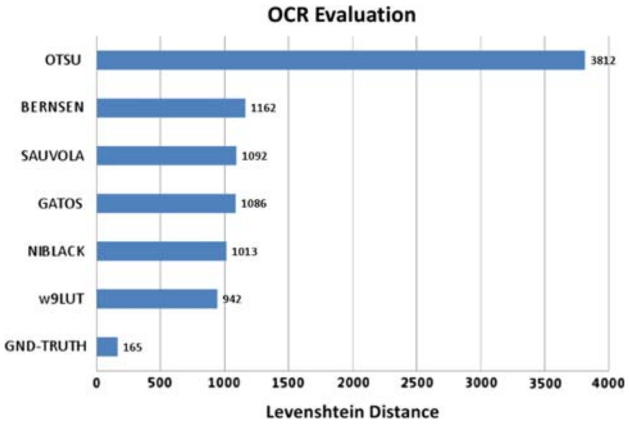


Fig. 15 OCR Accuracy evaluation using Levenshtein distance

yielded a better performance than usual because we prelabeled the text regions before applying the OCR engine. Thus, the background noise did not negatively impact it as much. Actually, it yielded a very good performance because its characters suffered much less broken character degradation. It is known that the Tesseract, as well as other OCR engines, perform very poorly at image resolution below 200 ppi. However, it is interesting to observe that 1) the LUT algorithm performs much better than the initial segmentation method it utilizes (Bernsen method), approximately 70% improvement in accuracy; and 2) the ground truth images yielded the best performance.

The experimental results demonstrate a 70% improvement in OCR result after enhancement of images with respect to results prior to LUT enhancement. The OCR performance of the ground truth images suggest that if we can improve on our LUT algorithm to yield outputs almost as good as the ground truth images, this will translate into an excellent OCR performance even at such a low resolution. However, OCR accuracy performance of each method varies from engine to engine. Our goal is to improve readability of the degraded documents not just to optimize performance of a particular OCR engine.

4.7.2 Global postprocessing techniques after binarization

In this section, we compare our proposed approach to known global postprocessing techniques carried out after binarization. The purpose is to show that the proposed system is more suitable for degradation correction than general postprocessing technique applied after binarization. Examples of known global postprocessing techniques include: mean filters, morphological operations, shrink and swell filtering, etc.

In the work done by Gatos et al. [20] on adaptive binarization of degraded documents, they utilized the postprocessing technique of a series of shrink and swell filtering operations in the final phase of their five-step process. The purpose was to eliminate the noise in the resulting binarized image obtained from segmentation, improve the quality of text regions and preserve stroke connectivity by isolated pixel removal and filling of possible breaks, gaps or holes. As observed in the test document image obtained through the Gatos et al. method in Fig. 14f, the shrink and swell filtering operations are suitable for small noise degradations such as shown in their experimental results [20] but are not sufficient to correct large degradations that our proposed LUT approach corrects. Their operation also requires proper fine tuning of many parameters per document image to obtain a quality result. In contrast, the LUT algorithm learns the pattern of degradation and correction using actual degraded data and does not involve fine tuning of different parameters at each stage. This is validated by the experimental results in Sect. 4.2.

The mean and median filters are usually used as a postprocessing technique to smooth image data, and eliminate noise [16]. Morphological operations such as erosion, dilation, open and close are also common postprocessing filters [30]. We applied three different filters: mean (5 × 5), erosion and dilation to segmented images and compared the results to the one obtained by our LUT filter. The morphological dilation operation was performed using a 3 × 3 rectangular structuring element while a 5 × 5 element was used for the erosion operation. As can be observed from Fig. 16, these postprocessing operations are unable to correct the large degradations in the images. This is because the degradation does not follow a simple model of small intrusions/protrusions. To obtain better quality results, the parameters for each operation need to be fine tuned and applied in an iterative adaptive manner. But this is a non-trivial task.

4.8 Evaluation of human perception of enhancement

Given that our goal is to improve the visual enhancement quality and readability of degraded document images, we carried out human evaluation of the enhanced images. We followed a similar evaluation scheme to that used in Kavallieratou et al. [22]. For their evaluation process, a human expert manually checked all the produced document images from the application of their enhancement system and was asked to compare these produced images with the original degraded image and classify each image in one of the following classes: 1) Better: the produced image has been

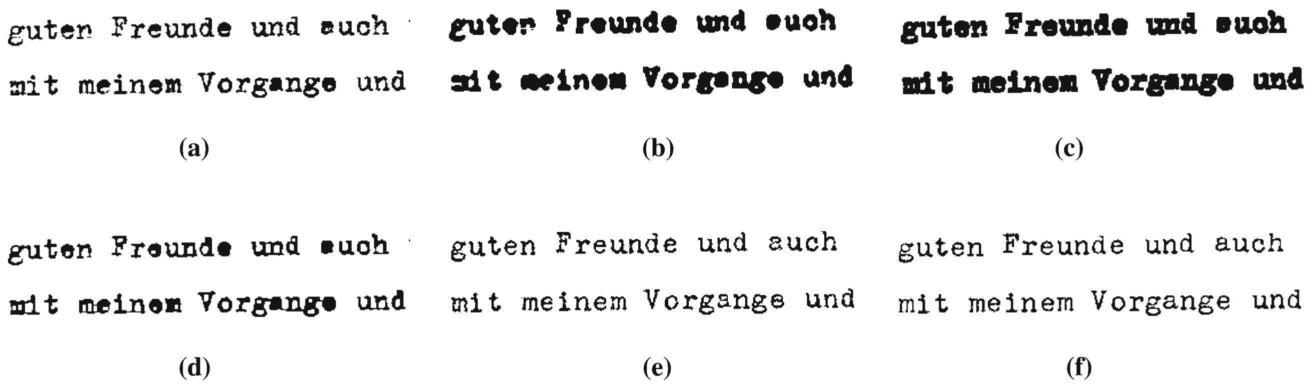

guten Freunde und auch
mit meinem Vorgange und
(a)
guten Freunde und auch
mit meinem Vorgange und
(b)
guten Freunde und auch
mit meinem Vorgange und
(c)
guten Freunde und auch
mit meinem Vorgange und
(d)
guten Freunde und auch
mit meinem Vorgange und
(e)
guten Freunde und auch
mit meinem Vorgange und
(f)

Fig. 16 Application of postprocessing techniques in comparison to proposed LUT method after initial segmentation using Bernsen method. **a** Binarization (BERNSEN), **b** mean filter, **c** dilation operation, **d** erosion operation, **e** proposed w-9LUT output image, **f** ground truth image

improved; 2) Same: the produced image and the degraded image are practically the same; 3) Worse: the produced image contains more noise than before.

When we carried out the evaluation initially by comparing entire document images pair as earlier, in all cases, the enhanced document image was perceived to be of better quality. We then repeated the experiment on individual characters. We randomly selected 100 characters each from a pair of binary degraded image and its enhanced image. We showed the human expert both characters from the same position from the image pair and asked them to classify which character image was better or if both were of same quality. To avoid a biased judgment, it is not revealed to the human expert which character was extracted from the degraded or enhanced image. We repeated this process for a subset of the document images enhanced using the w-9 LUT (and their corresponding binary degraded images). On the average, per 100 characters, 12% of the enhanced characters were judged to be of poorer quality, 20.5% of similar quality as their degraded counterparts and 67% of better quality. Thus, the human expert evaluation results validate the effectiveness of our proposed LUT algorithm in enhancement of degraded characters.

5 Conclusion

We have demonstrated the effectiveness of the LUT classifier system in learning the corrections of degradation patterns in historical typewritten document images. The main advantage of our method is that the degradation models is learned directly from the labeled data. Moreover, because the LUT system learns a non-linear degradation model from the data, any artifacts that may be introduced by the segmentation algorithm become part of the degradation model that needs to be estimated. We also show that the LUT outperforms other known state-of-the-art classifiers such as the SVM in its machine learning ability for this task.

The main limitation of our approach is that it is dependent on generating good quality ground truth images for the document collection to which it will be applied. The ground truthing process could be time consuming depending on the method used. However, we demonstrate that a very small set of expert-labeled data set is sufficient to train the LUT system. Thus, it makes the ground truthing process a worthwhile effort for large document collections. We have evaluated the performance of our system on the Frieder collection [17] which is a diverse collection containing documents in multiple languages written over a period of years. Our system can be applied to any other historical collection, as long as there is availability of a small subset of ground truth data for the collection.

The proposed approach can handle moderate skewness in the data collection. It is not uncommon for typewritten documents to be skewed of misaligned. Part of the document images tested had these features and the classifier was still able to correct the degradations in the text without altering the skewness or alignment of the characters. The LUT system is sensitive to the resolution of the images. For optimal results, one needs to ensure that the training dataset is of a similar resolution to the test collection. Finally, the proposed approach can only correct degradations for which it is trained.

We have also demonstrated that the effectiveness of the LUT classifier is further improved by arranging the LUT classifiers in a cascade configuration. In future work, we plan to combine the effectiveness of these classifiers using more complex ensembles of cascade configurations to improve performance. Our proposed system is an effective local similarity model that is efficient and can handle large feature descriptors. It utilizes a lookup table (LUT) classification algorithm in conjunction with nearest neighbor approximation to learn and correct patterns of degradation in a document image pixel by pixel. The key strength of the local similarity model is that it takes into account the context (neighborhood) of each pixel while making a classification decision without making general assumptions about dependence.

Acknowledgments The authors thank Dr. R. Haralick for initial discussions and suggestions concerning this work.

References

1. Agam, G., Bal, G., Frieder, G., Frieder, O.: Degraded document image enhancement. In: Lin, X., Yanikoglu, B.A. (eds.) Document Recognition and Retrieval XIV. Proceeding of the SPIE, vol. 6500, pp. 65000C–1–65000C–11 (2007)
2. Allier, B., Bali, N., Emptoz, H.: Automatic accurate broken character restoration for patrimonial documents. Int. J. Document Anal. Recognit. **8**(4), 246–261 (2006)
3. Antonacopoulos, A., Castilla, C.: Flexible text recovery from degraded typewritten historical documents. In: Proceedings of the 18th International Conference on Pattern Recognition ICPR'06, pp. 1062–1065 (2006)
4. Antonacopoulos, A., Karatzas, D.: A complete approach to the conversion of typewritten historical documents for digital archives. In: Proceedings of the IAPR International Workshop on Document Analysis Systems DAS'04, pp. 90–101 (2004)
5. Antonacopoulos, A., Karatzas, D.: Document image analysis for world war ii personal records. In: Proceedings of the International Workshop on Document Image Analysis for Libraries DIAL'04 (2004)
6. Antonacopoulos, A., Karatzas, D.: Semantics-based content extraction in typewritten historical documents. In: Proceedings of the International Conference on Document Analysis and Recognition ICDAR'05 (2005)
7. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching. J. ACM **45**(6), 891–923 (1998)
8. Badekas, E., Nikolaou, N., Papamarkos, N.: Text binarization in color documents. Int. J. Imaging Syst. Technol. **16**(6), 262–274 (2006)
9. Baird, H.: Document image quality: making fine discriminations. In: Proceedings of the International Conference on Document Analysis and Recognition (1999)
10. Bal, G., Agam, G., Frieder, G., Frieder, O.: Interactive degraded document enhancement and ground truth generation. In: Yanikoglu, B., Berkner, K. (eds.) Document Recognition and Retrieval XV. Proceedings of the SPIE, vol. 6815 (2008)
11. Bernsen, J.: Dynamic thresholding of gray-level images. In: Proceedings of the 8th International Conference on Pattern Recognition. pp. 1251–1255 (1986)
12. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**, 121–167 (1998)
13. Cannon, M., Hochberg, J., Kelly, P.: Quality assessment and restoration of typewritten document images. Int. J. Document Anal. Recognit. **2**(2–3), 80–89 (1999)
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines, (2001) Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
15. Droettboom, M., MacMillan, K., Fujinaga, I.: The gamera framework for building custom recognition systems. In: Symposium on Document Image Under standing Technologies. pp. 275–286 (2003)
16. Du, K., Lu, J., Sekiya, H., Sun, Y., Yahagi, T.: Postprocessing for restoring edges and removing artifacts of low bit rates wavelet-based image. In: Proceedings of the International Symposium on Intelligent Signal Processing and Communications, ISPACS '06, pp. 943–946 (2006)
17. The diaries of Rabbi Dr. Avraham Abba Frieder. http://ir.iit.edu/collections/
18. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. **3**(3), 209–226 (1977)
19. Gatos, B., Pratikakis, I., Perantonis, S.J.: An adaptive binarization technique for low quality historical documents. In: International Workshop Document Analysis Systems (DAS), pp. 102–113 (2004)
20. Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. Pattern Recognit. **39**(6), 317–327 (2006)
21. Kanungo, T.: Document Degradation models and a methodology for degradation model validation. Ph.D. thesis, University of Washington (1996)
22. Kavallieratou, E., Stamatatos, E.: Improving the quality of degraded document images. In: International Conference Document Image Analysis for Libraries DIAL'06 (2006)
23. Niblack, W.: An Introduction to Digital Image Processing. Prentice Hall (1986)
24. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
25. Obafemi-Ajayi, T., Agam, G., Frieder, O.: Ensemble lut classification for degraded document enhancement. In: Yanikoglu, B., Berkner, K. (eds.) Document Recognition and Retrieval XV. Proceedings of the SPIE, vol. 6815 (2008)
26. Sauvola, J., Pietikainen, M.: Adaptive document image binarization. Pattern Recognit. **33**, 225–236 (2000)
27. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative. J. Electron. Imaging **13**, 146–165 (2004)
28. Smith, R.: (2007) An overview of the tesseract ocr engine. In: Proceedings of the Int'l Conf. on Document Analysis and Recognition **2** 629–633
29. Stubberud, P., Kana, J., Kallurit, V. (1995) Adaptive image restoration of text images that contain touching or broken characters. In: Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)
30. Ye, X., Suen, C.Y., Cheriet, M.: A generic method of cleaning and enhancing handwritten data from business forms. Int. J. Document Anal. Recognit. **4**, 84–96 (2001)
31. Zheng, Q., Kanungo, T.: Estimation of morphological degradation model parameters. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '01. pp. 1961–1964 (2001)
32. Zheng, Q., Kanungo, T.: Morphological degradation models and their use in document image restoration. In: International Conference on Image Processing, pp.193–196 (2001)