LAMP-TR-
CAR-TR-
CS-TR-

October 2002

Machine Printed Text and Handwriting Identification in Noisy Document Images

Yefeng Zheng
Huiping Li
David Doermann

Machine Printed Text and Handwriting Identification in Noisy Document Images

Yefeng Zheng
Huiping Li
David Doermann

Language and Media Processing Laboratory
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275
(zhengyf,huiping,doermann)@cfar.umd.edu

Abstract

In this paper we address the problem of the identification of text from noisy documents. We especially segment and identify handwriting from machine printed text since 1) handwriting in a document often indicates corrections, additions or other supplemental information that should be treated differently from the main or body content, and 2) the segmentation and recognition techniques for machine printed text and handwriting are significantly different. Our novelty is that we treat noise as a distinguish class and model noise based on selected features. Trained Fisher classifiers are used to identify machine printed text and handwriting from noise. We further exploit context to refine the classification. A Markov Random Field (MRF) based approach is used to model the geometrical structure of the printed text, handwriting and noise to rectify the misclassification. Experimental results show our approach is promising and robust, and can significantly improve the page segmentation results in noise documents.

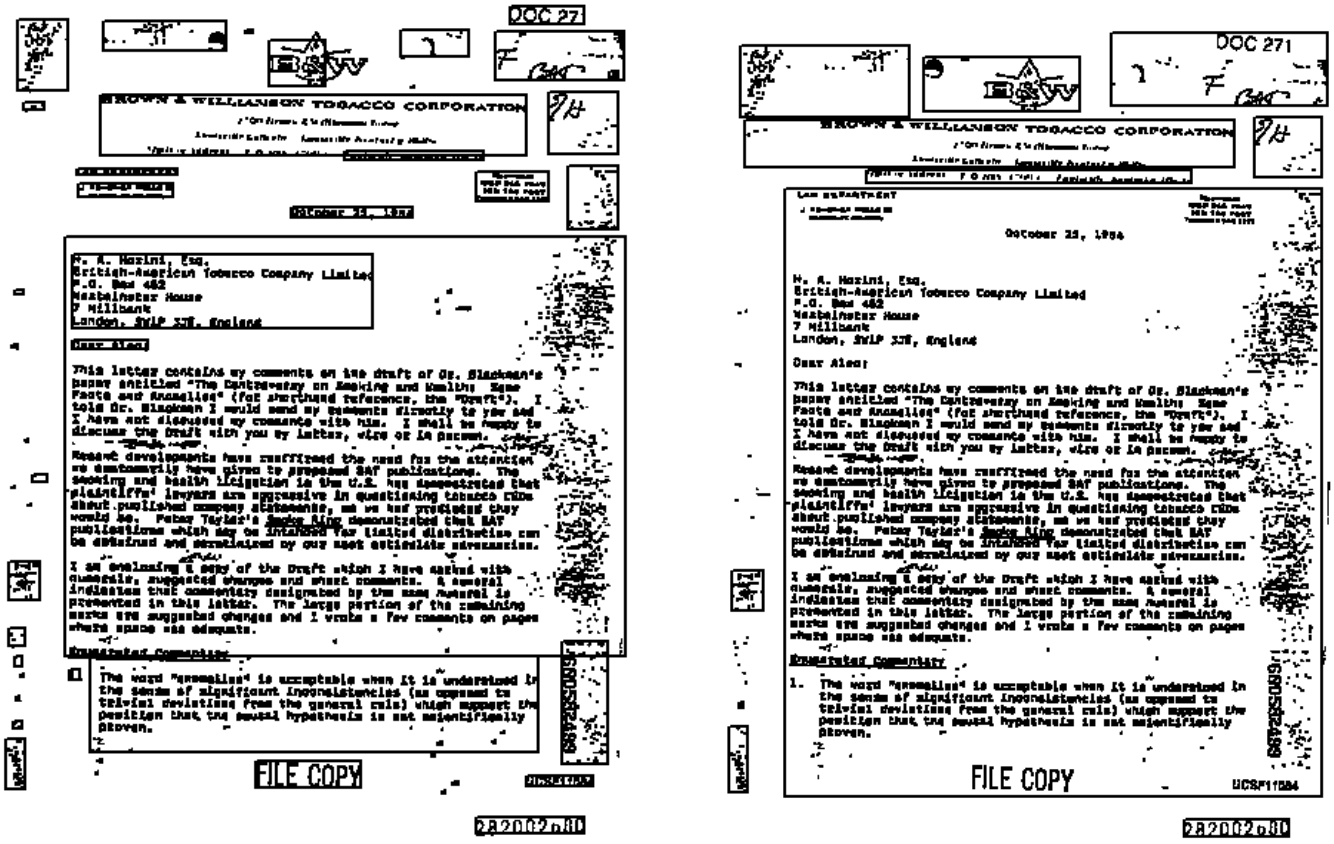
Keywords: Text Identification, Handwriting Identification, Markov Random Field, Post-Processing, Noisy Document Image Enhancement, Document Analysis

1 Introduction

Documents are the results of a set of physical process and conditions. The resulting document can be viewed as layers: letter head, content, signatures, annotations, noise, etc. Document analysis reverses such process to segment document into layers with different physical and semantic properties. After decades of researches, automatic document analysis has advanced to a point that text segmentation and recognition has been viewed as a solved problem in clean, well-constrained documents. However, the performance degrades quickly even when a small amount of noise is introduced. For example, a typical bottom-up page segmentation method starts from the extraction of connected components [24, 45]. Based on the spatial proximity and size, connected components are then merged as text lines and zones. A classification process is further used to identify zone types (text, table, images, etc.). These algorithms work well on clean documents where zones with different properties are well separated. However, they often fail on noisy documents where noise often mixes with and/or spatially close to text regions. As an example, Figure 1(a) and (b) show the segmentation results of an extremely noisy document when we use Docstrum algorithm [45] and ScanSoft SDK [1]. Text and noise are erroneously segmented into the same zones for both algorithms.

In this paper we present a novel approach to identify text from extremely noisy documents. Instead of simple noise filtering used in other work [24, 45], we treat noise as a distinguish class and model it based on selected features. We further identify handwriting from machine printed text since: 1) most handwriting in a document often indicates corrections, additions or other supplemental information that should be treated differently from the main or body content, and 2) the segmentation and recognition techniques for machine printed text and handwriting are significantly different. Based on these considerations, we treat the problem as a three-class (machine printed text, handwriting and noise) identification problem.

In practice mis-classification often happens due to the overlapping in the feature space. This is especially true between handwriting and noise. To deal with this problem, we exploit contextual information as a post-processing to refine the classification. Contextual



(a) (b)

Figure 1: Page segmentation results of an extremely noisy document using Docstrum algorithm and ScanSoft SDK. Noise is segmented into text zone erroneously in both cases. (a) Docstrum, (b) ScanSoft.

information is very useful to improve the classification accuracy. It is widely used in many OCR systems and its effectiveness has been demonstrated in previous work [22, 49]. The key is to model the statistical dependency among neighboring components. The output of an OCR system is a text stream which is one dimensional. Therefore, an N-gram language model, based on an N-order 1-D Markov chain, is effective to model the context. With assistance from a dictionary, the N-gram approach can correct most recognition errors. Images are two dimensional. Generally, 2-D signals are not causal, and it is much harder to model the dependency among neighboring components in an image. Among image models studied so far, Markov Random Field (MRF) is widely studied and successfully used in many applications. MRF is suitable for image analysis because the local statistical dependency of an image can be well modeled as Markov properties. It can incorporate *a priori* contextual information or constraints in a quantitative way. MRF model has been extensively used in various image analysis applications such as texture synthesis and segmentation, edge detection, and image restoration [17, 36]. In this paper, we use MRF to model the dependency of segmented neighboring blocks.

The diagram of our system is shown in Figure 2. The documents we are processing are extremely noisy with machine printed text, handwriting and noise mixed. We first extract the connected components and merge them at the word level based on the spatial proximity. We then extract several categories of features and use trained Fisher classifiers to classify each word into machine printed text, handwriting or noise. Finally, contextual information is incorporated into MRF models to refine the classification results further.

The rest of the paper is organized as follows: Section 2 is a literature survey of the related work, followed by the detailed description of our classification method in Section 3. MRF based post-processing is presented in Section 4, and the experimental results are presented in Section 5. The paper is ended with some discussions and future work.

2 Related Work

The research presented in this paper is related to previous work on page segmentation, zone classification, handwriting identification, and document enhancement.

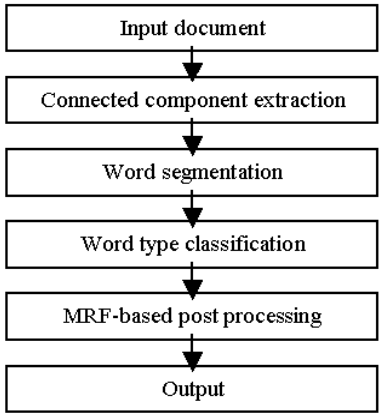


Figure 2: The diagram of our system.

2.1 Page Segmentation

Previous work on page segmentation can be broadly classified into three categories: bottom-up [24, 45], top-down [42], and hybrid [54]. In a typical bottom-up approach such as Docstrum algorithm proposed by O'Gorman [45], connected components are extracted first, then merged into words, lines, zones and columns hierarchically based on the size and spatial proximity. Bottom-up methods can handle documents with complex layouts. However, it is time consuming, and sensitive to noise.

On the other hand, a typical top-down method, such as X-Y cut proposed by Nagy [42], starts from the whole document and splits it recursively into columns, zones, lines, words and characters. A top-down method is effective for the documents with regular layout, but fails when documents have a non-Manhattan structure.

Another problem for X-Y cut is the global parameters for the optimal segmentation is often difficult to achieve if pre-knowledge is not available. Sylwester et al. proposed a hybrid method which starts from the top [54]. First, they over-segment a document into small zones with more conservative thresholds. Then they use the bottom-up method which groups over-segmented small zones with the same properties into a single zone.

All of the above methods are based on the analysis of foreground (black pixels). As an alternative, white stream methods based on the analysis of background (white pixels) are presented in [6, 47]. In these methods, rectangles covering the white gap (white pixels) between foreground are extracted. Foreground regions surrounded by these white

rectangles are extracted as zones. A more comprehensive survey is presented in [20].

2.2 Zone Classification

Zone classification labels the content of each segmented zone as one of the pre-defined type [24, 47, 56], such as text, image, figure and table. Pavlidis et al. use the correlation of horizontal scan lines and black pixel density as features to classify each zone into text, diagram and half tone images [47]. The basic idea is the correlation, $C(r, y)$, of text region between scan line y and $y + r$ is different from that of a half tone image region. For a text region, the correlation of neighboring scan lines are high, but decreases quickly when the distance between two scan lines increases. For example, a text region has high average $C(1, y)$ and low average $C(5, y)$. For half tone images, to avoid the appearance of regular patterns, both $C(1, y)$ and $C(5, y)$ are small and roughly the same. The black pixel density is used to further distinguish diagram from text. Wang et al. used 69 features, such as run length mean and variance, spatial mean and variance, fraction of the total number of black pixels in the zone, the zone width ratio of each zone, and the number of text glyphs in the given zone, to classify each zone into 9 classes. They did experiments on ground-truthed zones of UW III database, and achieved the accuracy as high as 98.52% [56]. Jain et al. directly perform classification on the generalized lines (GTLs) extracted using a bottom-up approach [24]. If the height of a GTL is less than a threshold and the connected components in it are horizontally aligned, it is classified as a text line. Text lines and non-text lines are merged into text regions and non-text regions respectively. They further classify non-text regions into images, tables and drawings. If the minimal height and width of all GTLs inside a non-text region is larger than a threshold and the black pixel density is large enough, it is classified as an image region. The regions containing two parallel top and bottom horizontal lines are classified as tables. The remaining non-text regions are classified as drawings. It works well for long text lines, but may fail when text lines are short.

Some other approaches treat text, images and figures as different textures, and use trained classifiers to segment and identify them [11, 23, 33]. They often directly work on gray scale images, and need classification of each pixel. To reduce the computation

complexity, multi-resolution based techniques are often used.

2.3 Handwriting Identification

Some work has been done on handwriting/machine printed text identification. The classification is typically performed at the text line [13, 14, 46, 51], word [19], or character level [31, 61]. At the line level, machine printed text lines are typically arranged regularly, while handwritten text lines are irregular. Srihari et al. implemented a text line based approach using this characteristic and achieved the classification accuracy of 95% [51]. One advantage of the approach is it can be used in different scripts (Chinese, English etc.) with little or no modification. Guo et al. proposed an approach based on the vertical projection profile of the word [19]. They used a Hidden Markov Model (HMM) as the classifier and achieved the classification accuracy of 97.2%. Although at the character level less information is available, humans can still identify the handwritten and machine printed characters easily, inspiring researchers to pursue classification at the character level. Kuhnke proposed a neural network-based approach with straightness and symmetry as features [31]. Zheng et al. used run-length histogram features to identify handwritten and printed Chinese characters and achieved promising results [61]. In previous work, we implemented a handwriting identification method based on several categories of features and a trained Fisher classifier [60]. However, the noise problem is not addressed.

2.4 Document Enhancement

There are two types of degradation in document images: 1) The physical degradation of papers of documents, and 2) degradation introduced by digitalization. Both of them will deteriorate the performance of a document analysis system significantly, if severe enough. Several document degradation models [4, 28, 53], document quality assessment [7, 34], and document enhancement algorithms [12, 21, 28, 35, 38, 40, 43, 44, 59] have been presented in previous work. One common enhancement approach is window-based morphological filtering [38, 40, 44]. Morphological filtering performs a look up table procedure to determinate the output of ON (black pixel) or OFF (white pixel) for each entry of

the table, based on a windowed observation of its neighbors. These algorithms can be further categorized as manually designed, semi-manually designed or automatically trained approaches. kFill algorithm, proposed by O'Gorman [44], is a manually designed approach and has been used by several other researchers [7, 8]. It considers a $k \times k$ window which comprises an inside $(k-2) \times (k-2)$ region called the *core*, and the $4(k-1)$ pixels on the window perimeter called the *neighborhood*. The filling operation sets all values of the core to ON or OFF, depending upon the pixel values in the neighborhood. Experiments show it is effective to remove salt-and-pepper noise. Liang et al. proposed a semi-manually designed approach with a 3×3 window size [37]. They manually determine some entries to output ON or OFF based on *a priori* observation. The remaining entries are trained to select the optimal output. It is difficult to manually design a filter with a large window size, and the success depends on the experience. If both the ideal and degraded images are available, optimal filters can be designed by training [40]. After registering the ideal and degraded images at the pixel level, an optimal look-up table, based on the number of a specific windowed context outputs, can be designed. However it is difficult to train, store and retrieve the look-up table when the window size is large. Two approaches are used to solve these problems: 1) using specially structured filters (*increasing filters*) to approximate the optimal filters, and 2) using iterative filtering with a small window size to approximate the performance of a large window size [40, 58]. Though it is a general method and optimal in statistical sense, this approach requires both the original and the corresponding degraded images for training. Loce used artificially degraded images generated by models for training, while Kanungo et al. proposed methods for the validation and parameter estimation of degradation models [29, 27, 30]. Though the uniformity and sensitivity of his approach is tested by some other researchers [5, 53], no degradation model has been declared to pass the validation. Another problem of morphological approaches is the small window size. The most commonly used window size is 5×5, which is too small to contain enough information for enhancement. Other methods, such as averaging over multiple video frames [35] or multiple instances in the same document page [21], marginal noise removal [12], show through removal [43, 55], and geometric distortion calibration [28, 59], are proposed to deal with various types of

degradation.

Ideally the image quality should be estimated first so the corresponding enhancement algorithms can be applied automatically. Cannon et al. proposed a document quality assessment algorithm based on five factors: small speckle factor, white speckle factor, touching character factor, broken character factor, and font size factor [7]. They used a linear classifier to select the best one out of four enhancement algorithms, and can reduce the OCR error rate from 20.27% to 12.60% on their database. Li el al. proposed an approach for the quality estimation of color video text, which classifies the video text quality into 6 levels [34]. Besides these global approaches, some researchers select different enhancement algorithms for each small region in an image [3, 52].

Most of above approaches are focused on improving the OCR accuracy in noisy documents. As shown in Figure 1, degradation will not only deteriorate OCR performance, but other document processing tasks, such as page segmentation. Few work has been addressed in this area. The difference between our approach and the previous work is that we perform noise identification at the word level, which contains more information than a small window. Furthermore, contextual information of neighboring words are exploited for post-processing to refine the identification. Experiments show our noise removal algorithm can increase the page segmentation accuracy significantly.

3 Text Identification

In this section we present our text (machine printed or handwritten) extraction and classification method.

3.1 Pre-processing

A special consideration must be given to the size of the region being segmented before we can perform any classification. If the region is too small, the information contained in it may be not sufficient for classification; if the region is too large, however, different types of components may be mixed in the same region. In previous work we conducted a performance evaluation for the classification accuracy of machine printed text and

handwriting at the character, word and zone levels [60]. It shows a reliable classification can be achieved at the word level. Therefore, we segment images into the word level and then perform classification. Since noise has no concept of *word* , we use terminology *block* and *word* interchangeably in the following presentations.

We first extract connected components, and then merge them into words based on the geometric proximity and size. Those connected components with extremely large size or extremely large or small aspect ratio are filtered out directly. However, noise with the similar size as text can not be filtered out. To identify text from this type of noise is our focus.

3.2 Feature Extraction

Several sets of features are extracted for classification. The description and dimension of each feature set is listed in Table 1. Machine printed text, handwriting and noise have different visual appearance and physical structures. Therefore, structural features are extracted to reflect this difference. Gabor filter features and run-length histogram features can capture the difference in stroke orientation and stroke length between handwriting and printed text. Compared with text, noise blocks often have simple stroke complexity. Therefore, crossing counts histogram features are exploited to model such difference. We further take regions of machine printed text, handwriting and noise blocks as different textures. Two sets of bi-level texture features (bi-level co-occurrence features and bi-level 2×2-gram features) are used for classification. In the following subsections we present these features in detail.

3.2.1 Structural Features

We extract two sets of structural features. The first set includes features related to the physical size of the blocks such as the density of black pixels, the width, the height, the aspect ratio, and the area. Suppose the image of the block is $I(x, y)$, $0 \leq x < w$, $0 \leq y < h$, and w, h are the width and height respectively. Each pixel in the block has two values: 0 representing background (white pixel) and 1 representing content (black

pixel). Then the density of the black pixels d is:

$$d = \frac{\sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x,y)}{w \times h} \tag{1}$$

The size of machine printed words is more consistent than that of handwriting and noise on the same page. However, machine printed words on different pages may vary significantly. Therefore, we use a histogram technique to estimate the dominant font size [45], and then use the dominant font size to normalize the width, height, aspect ratio and area of the block respectively. Suppose the estimated dominant character width and height are w_d and h_d respectively, then the normalized features, w', h', p' and a', are:

$$w' = w/w_d \tag{2}$$

$$h' = h/h_d \tag{3}$$

$$p' = \frac{w'}{h'} \tag{4}$$

$$a' = w' \times h' \tag{5}$$

The second set of structural features are based on the connected components inside the block, such as the mean and variance of the width (m_w and σ_w), height (m_h and σ_h), aspect ratio (m_p and σ_p), and area (m_a and σ_a) of connected components. The sizes of connected components inside a machine printed word are more consistent, leading to smaller σ_w and σ_h. For a handwritten word or noise block, the bounding boxes of the connected components tend to overlap with each other, resulting in a larger overlapping rate (Figure 3). For machine printed English words, however, each character forms a connected component not overlapping with others. The overlapping area (the summation of areas of the gray rectangles in Figure 3) normalized by the total area of the block is calculated as a feature. Another feature we use is the variance of the vertical projection. In a machine printed text block, the vertical projection profile has obvious valleys and peaks since neighboring characters do not touch each other. However, for a handwritten word or noise block, the vertical projections are much smoother, resulting in smaller variance.

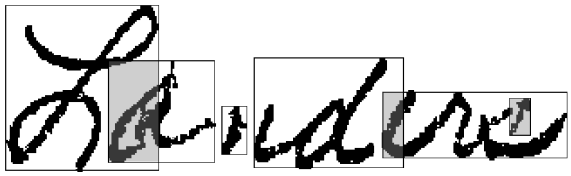


Figure 3: The overlap of the connected components inside a handwritten word.

3.2.2 Gabor Filter Features

Gabor filter can represent signals in both the frequency and time domains with minimum uncertainty [16] and has been widely used for texture analysis and segmentation [23]. Researchers found that it matches the mammal's visual system very well, which provides further evidence that we can use it in our classification tasks. In spatial and frequent space, the two dimensional Gabor filter is defined as:

$$g(x,y) = \exp\left\{-\pi\left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right]\right\} \times \cos\{2\pi(u_0 x + v_0 y)\} \tag{6}$$

$$\begin{aligned} G(u,v) = 2\pi\sigma_x\sigma_y(exp\{-\pi[(u'-u_0')^2\sigma_x^2 + (v'-v_0')^2\sigma_y^2]\} \quad + \\ exp\{-\pi[(u'+u_0')^2\sigma_x^2 + (v'+v_0')^2\sigma_y^2]\}) \end{aligned} \tag{7}$$

where $x' = -x\sin\theta + y\cos\theta$, $y' = -x\cos\theta - y\sin\theta$, $u' = u\sin\theta - v\cos\theta$, $v' = -u\cos\theta - v\sin\theta$, $u_0' = -u_0\sin\theta + v_0\cos\theta$, $v_0' = -u_0\cos\theta - v_0\sin\theta$, $u_0 = f\cos\theta$, and $v_0 = f\sin\theta$. Here f and θ are two parameters, representing the central frequency and orientation of the Gabor filter.

Suppose an original image is $I(x,y)$, then the filtered image $I_k'(x,y)$ using the kth Gabor filter $g_k(x,y)$ can be described as:

$$I_k'(x,y) = I(x,y) * g_k(x,y) \tag{8}$$

It is very expensive, however, to calculate the filter in the spatial domain defined in Equation 8. Instead, we use an FFT to calculate it in the frequent domain. Let $I(u,v)$ be the FFT transform of the original image and $G_k(u,v)$ be the frequency response of the kth Gabor filter. Then the frequency spectrum of filtered image $I_k'(u,v)$ equals to the product of $I(u,v)$ and $G_k(u,v)$:

$$I_k'(u,v) = I(u,v)G_k(u,v) \tag{9}$$

The filtered image can be achieved by calculating the inverse FFT of $I'_k(u,v)$. For Gabor filters with different parameters, $G_k(u,v)$ is calculated and pre-stored. $I(u,v)$ is calculated only once and shared among different Gabor filters. This can reduce the computation significantly. The variance of the filtered image, F_k, $k = 1, 2, ...N$, are taken as features:

$$F_k = \sqrt{\frac{1}{w \times h} \sum_x \sum_y (I_k(x,y) - m_k)^2} \tag{10}$$

where

$$m_k = \frac{1}{w \times h} \sum_x \sum_y I_k(x,y) \tag{11}$$

In our experiments we let $\theta_k = k \times 180/N, k = 1, 2, \ldots N$, with $N = 16$. Altogether there are 16 Gabor filters, which generate 16 features.

3.2.3 Run-length Histogram Features

Run-length histogram feature is proposed by Zheng et al. for machine printed/ handwritten Chinese character classification [61]. These features are used in our case to capture the difference between the stroke length of machine printed text, handwriting and noise blocks. First, black pixel run-length of four directions, including horizontal, vertical, major diagonal and minor diagonal, are extracted. We then calculate four histograms of run-lengths for four directions respectively, as shown in Figure 4. To get scale-invariant features, we normalize the histogram first. Suppose C_k, $k = 1, 2, ..., N$ is the number of run-length with length k, and N is the maximal length for all possible run-length, then the normalized histogram C'_k is:

$$C'_k = \frac{C_k}{\sum_{i=1}^{N} C_i} \tag{12}$$

We then divide the histogram into five bins with equal width and use five Gaussian-shaped weight windows to get the final features (Figure 4). Taking horizontal run-length histogram as an example, the run-length histogram feature Rh_i is calculated as:

$$Rh_i = \sum_{k=1}^{w} G(k; u_i, \sigma) C'_k, \quad i = 1, 2, 3, 4, 5 \tag{13}$$

Where w is the width of the block, the maximal possible length of the horizontal run-length, u_i is the center of each weight window $u_i = (i - 0.5) \times w$, $\sigma = \frac{w}{10\sqrt{2\ln 2}}$, and

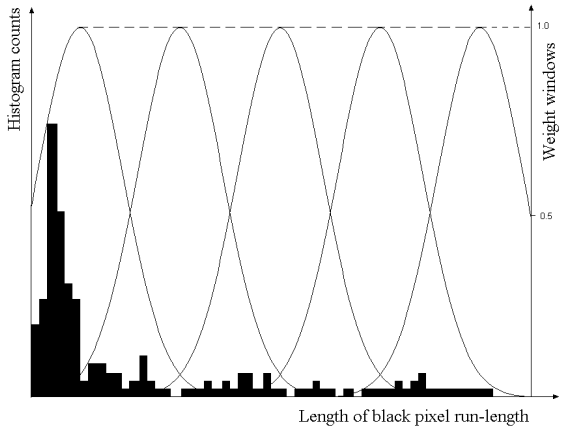


Figure 4: Run-length histogram features

$G(k; u_i, \sigma)$ is a Gaussian-shaped function:

$$G(k; u_i, \sigma) = exp\left\{-\frac{(k-u_i)^2}{2\sigma^2}\right\} \tag{14}$$

σ is chosen so the weight on each bin border is 0.5. Another alternative window is to use rectangles without overlap between neighboring bins. Experiments show the extracted features with Gaussian-shaped weight windows are more robust. 5 features are extracted in each direction, leading to 20 features.

3.2.4 Crossing Counts Histogram Features

A crossing count is the number of times the pixel value turns from 0 (white pixel) to 1 (black pixel) along horizontal or vertical raster scan lines. Crossing counts can be used to measure stroke complexity [2, 60]. In our approach, first, crossing counts for each horizontal and vertical scan line is calculated. Similarly we get two histograms. The same techniques are exploited to get final features from the histograms. Totally, 10 features are extracted.

3.2.5 Bi-level Co-occurrence Features

A co-occurrence count is the number of times a given pair of pixels occurs at a fixed distance and orientation. In the case of binary images, the possible occurrences are white-white, black-white, white-black and black-black at each distance and orientation.

In our case, we are concerned primarily with the foreground. Since the white background region often accounts up to 80% of a document page, the occurrence frequency of white-white or white-black pixel pairs would always be much higher than that of black-black pairs. The statistics of black-black pairs carry most of the information. To eliminate the redundancy and reduce the effects of over-emphasizing the background, we only consider black-black pairs. Four different orientations (horizontal, vertical, major diagonal and minor diagonal) and four distance levels (1, 2, 4, 8 pixels) are used for classification (altogether 16 features). The details can be found in [10].

3.2.6 Bi-level 2×2-gram Features

The N×M-gram was introduced by Soffer in the context of image classification and retrieval [50]. We use bi-level 2×2-grams at a hierarchy of distance from the origin. We first remove the dominant background (the all white gram), then scale each entry by multiplying the number of occurrence by a coefficient proportional to the number of black pixels in a 2×2-gram. The more black pixels, the larger the coefficient. In this work, we used $d^b(1-d)^{(4-d)}$, where d is the density of the image block, and b is the number of 1's in the 2×2-gram. We then normalize the entire vector of occurrences by dividing them by the sum of all occurrences. Four distances (1, 2, 4, 8 pixels) are chosen to extract features, generating 60 features. The details can be found in [10].

3.3 Feature Selection

As shown in Table 1 we totally extract 140 features from segmented blocks. When the number of training samples are limited, using a big feature set may decrease the generality of the classifier [15]. The bigger the feature set, the more training samples needed and the more expensive for feature extraction and classification. Therefore, we perform feature selection before feeding them to the classifier. We use forward search feature selection technique to conduct feature selection [32, 25]. We first divide the whole feature set $\mathcal{F}$ into currently selected feature set $\mathcal{F}_{\mathrm{s}}$ and un-selected feature set $\mathcal{F}_{\mathrm{n}}$, which satisfy:

$$\mathcal{F}_{\mathrm{s}} \cap \mathcal{F}_{\mathrm{n}} = \Phi \tag{15}$$

Table 1: Features used for machine printed text/handwriting/noise classification

Feature set	Feature description	# of features	# of features selected
Structural	Region size, connected components	18	9
Gabor filter	Stroke orientation	16	4
Run-length histogram	Stroke length	20	5
Crossing counts histogram	Stroke complexity	10	6
Bi-level co-occurrence	Texture	16	2
2×2 gram	Texture	60	5
Total		140	31

$$\mathcal{F}_{\mathrm{s}} \cup \mathcal{F}_{\mathrm{n}} = \mathcal{F} \tag{16}$$

Then, the selection procedure can be described as :

1. Set $\mathcal{F}_{\mathrm{s}} = \Phi$, and $\mathcal{F}_{\mathrm{n}} = \mathcal{F}$.
2. Label all features in $\mathcal{F}_{\mathrm{n}}$ as un-tested.
3. Select one un-tested feature $f \in \mathcal{F}_{\mathrm{n}}$ and label it as tested.
4. Put f and $\mathcal{F}_{\mathrm{s}}$ together, and generate a temporary selected feature set $\mathcal{F}_{\mathrm{s}}^{\mathrm{f}}$.
5. Estimate the classification accuracy with feature set $\mathcal{F}_{\mathrm{s}}^{\mathrm{f}}$ using 1-NN classifier and leave-one-out cross validation technique. The basic idea is at each iteration, only one sample is used for testing, while others are used for training. We repeat this process until all samples are used as testing samples once. The average accuracy for all iterations is taken as the estimated accuracy for the current feature set. Leave-one-out cross validation technique can estimate the accuracy of a classifier with small variation [15].
6. If there are un-tested features in $\mathcal{F}_{\mathrm{n}}$, goto step 3.
7. Find a feature $\hat{f} \in \mathcal{F}_{\mathrm{n}}$, such that the corresponding temporary feature set $\mathcal{F}_{\mathrm{s}}^{\mathrm{f}}$ has the highest classification accuracy:

$$\hat{f} = arg \max_{f \in \mathcal{F}_{\mathrm{n}}} Accuracy(\mathcal{F}_{\mathrm{s}}^{\mathrm{f}}) \tag{17}$$

the remove $\hat{f}$ from $\mathcal{F}_{\mathrm{n}}$ to $\mathcal{F}_{\mathrm{s}}$.

8. If $\mathcal{F}_{\mathrm{n}} \neq \Phi$, go to step 2; otherwise exit.

We use LNKnet pattern classification software to conduct our feature selection experiments [32]. LNKnet provides several classifiers, such as likelihood classifiers, k-NN classifier, Neural Network classifiers, etc., and several feature selection algorithms such as forward search, backward search, and forward and backward search. We select 1-NN classifier and forward search feature selection algorithm for our experiments based on the following considerations:

1. Feature selection is an extremely expensive task. For the forward search algorithm we used, the number of feature set evaluated is $140 + 139 + \cdots + 1 = 9,870$. Backward search algorithm has similar performance as forward search algorithm and even more expensive [25]. Forward and backward search algorithm has better performance than forward search algorithm, however, the number of feature set evaluated is much larger.

2. To use leave-one-out cross validation to estimate the accuracy of a feature set, we need to rotate each sample as the testing sample once. Therefore, the number of classifiers trained is the number of all samples. In our experiments, there are about 4,500 samples. Considering the big number of feature sets needed for evaluation, we choose lightweight classifier such as k-NN classifier.

We collect about 1,500 blocks in each class in our feature selection experiments. As shown in Figure 5, when the number of selected features increases the mis-classification rate decreases sharply at first. The trend reverses after some point. The best classification is achieved when only 31 features are selected, with an error rate of 5.7%. When all features are used, the mis-classification rate increases to 9.2% due to the limited number of training samples and large feature set. The last column in Table 1 lists the number of features selected in each set. It shows that texture features, such as bi-level co-occurrence and 2×2 grams, are less discriminant than other categories, mainly due to the small region size. Only 1/8 of bi-level co-occurrence features and 1/12 of 2×2 gram features are

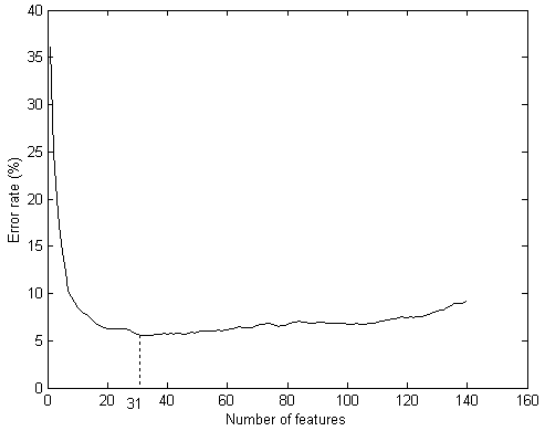


Figure 5: Feature selection experiments. The best classification result is achieved when 31 features are selected.

selected. Crossing counts histogram features and structural features are very effective, with more than half of the original features in both sets selected in the final feature set.

3.4 Classification

Compared with Neural Network (NN) and Support Vector Machine (SVM), Fisher classifier is easier to train, faster for classification and needs fewer training samples. Furthermore, it will not cause over-training problems. We use Fisher classifier for classification. For a feature vector $\underline{X}$, the Fisher classifier projects $\underline{X}$ onto one dimension Y in direction $\underline{W}$:

$$Y = \underline{W}^T \underline{X} \tag{18}$$

The Fisher criterion finds the optimal projection direction $\underline{W}_o$ by maximizing the ratio of the between-class scatter to the within-class scatter, which benefits the classification. Let $\underline{S}_w$ and $\underline{S}_b$ be within- and between-class scatter matrix respectively,

$$\underline{S}_w = \sum_{k=1}^{K} \sum_{\underline{x} \in \text{class } k} [(\underline{x} - \underline{u}_k)(\underline{x} - \underline{u}_k)^T] \tag{19}$$

$$\underline{S}_b = \sum_{k=1}^{K} (\underline{u}_k - \underline{u}_0)(\underline{u}_k - \underline{u}_0)^T \tag{20}$$

$$\underline{u}_0 = \frac{1}{K} \sum_{k=1}^{K} \underline{u}_k \tag{21}$$

where $\underline{u}_k$ is the mean vector of the kth class, $\underline{u}_0$ is the global mean vector and K is the number of the classes. The optimal projection direction is the eigenvector of $\underline{S}_w^{-1}\underline{S}_b$ corresponding to the largest eigenvalue [15]. For a two-class classification problem, we do not need to calculate the eigenvector of $\underline{S}_w^{-1}\underline{S}_b$. It is shown that the optimal projection direction is:

$$\underline{W}_o = \underline{S}_w^{-1}(\underline{u}_1 - \underline{u}_2) \tag{22}$$

Let Y_1 and Y_2 be the projection of two classes and $\mathrm{E}[Y_1]$ and $\mathrm{E}[Y_2]$ be the mean of Y_1 and Y_2 respectively. Suppose $\mathrm{E}[Y_1] > \mathrm{E}[Y_2]$, then the decision can be made as:

$$C(\underline{x}) = \begin{cases} \text{class 1} & \text{If } y > (\mathrm{E}[Y_1] + \mathrm{E}[Y_2])/2 \\ \text{class 2} & \text{Otherwise} \end{cases} \tag{23}$$

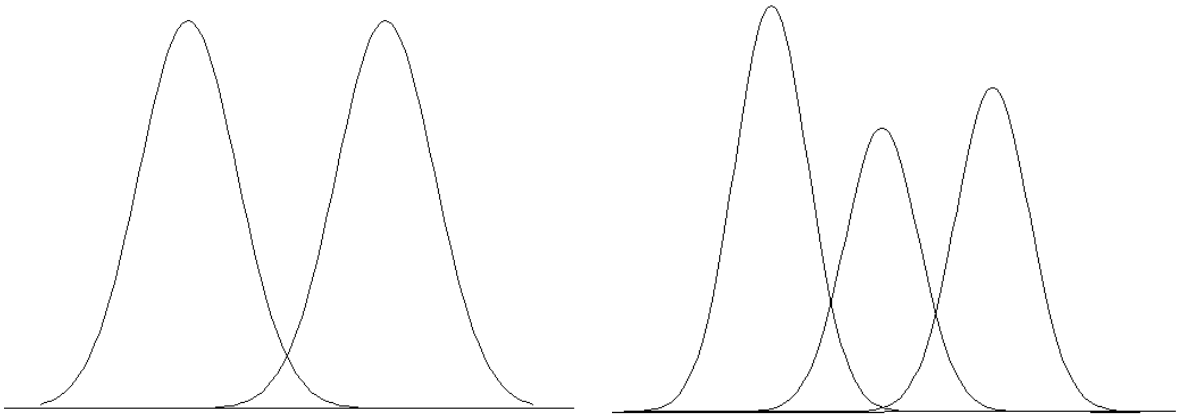
It is shown that if the feature vector $\underline{X}$ is jointly Gaussian distributed, and two classes have the same covariance matrices, then the Fisher classifier is optimal in a minimum classification error sense [15].

Fisher classifier is often used for two-class classification problems. Although it can be extended to multi-class classification (three classes in our case), the classification accuracy decreases due to the overlap between neighboring classes (Figure 6(b)). Therefore, we use three Fisher classifiers, each optimized for two-class classification (machine printed text/handwriting, machine printed text/noise and handwriting/noise), respectively. Each classifier outputs a confidence of the classification and the final decision is made by fusing the outputs of all three classifiers.

3.5 Classification Confidence

In Fisher classifier, the feature vector is projected onto an axis, on which the ratio of between-class scatter to within-class scatter is maximized, as shown in Figure 6(a). According to the central limit theorem, the distribution of the projection can be approximated by a Gauss distribution, if no feature has dominant variance over others, as follows [18]:

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{y-m}{\sigma}\right)^2\right] \tag{24}$$



(a) (b)

Figure 6: Projection of features in Fisher classifier. (a) Two classes, (b) three classes.

Where $f_Y(y)$ is the probability density function of the projection. The parameters, m and σ can be estimated from training samples. Classification confidence $C_{i,j}$ of class i using classifier j is defined as:

$$C_{i,j} = \begin{cases} \frac{f_Y(y/\underline{X} \in \text{class i})}{f_Y(y/\underline{X} \in \text{class i}) + f_Y(y/\underline{X} \in \text{another class})} & \text{If } i \text{ is applicable for classifier } j. \\ 0 & \text{Otherwise} \end{cases} \tag{25}$$

Where i is the class label, and j represents the trained classifiers. The final classification confidence is defined as:

$$C_i = \frac{1}{2} \sum_{j=1}^{3} C_{i,j} \tag{26}$$

$C_{i,j} \in [0, 1]$ for two applicable classifiers and $C_{i,j} = 0$ for the third classifier. So $C_i \in [0, 1]$. However, C_i is not a good estimation of the *post-priori* probability since $\sum_{i=1}^{3} C_i = 1.5$, instead of 1. We can take C_i as an estimation of a non-decreasing function of the *post-priori* probability, which is a kind of generalized classification confidence [39].

Figure 7 show the word segmentation and classification results for the whole and parts of a document image, with blue, red and green rectangles representing machine printed text, handwriting and noise respectively. We can see most of the blocks are correctly classified. However some blocks are mis-classified due to the overlapping in the feature space. For example, some noise blocks are classified as handwriting in Figure 7(b), and some small printed words are classified as noise in Figure 7(c). In next section,

we present a MRF based post-processing to refine the classification by incorporating contextual information.

4 MRF Based Post-Processing

4.1 Background

Let $\underline{X}$ denotes the random field defined on Ω and Γ denotes the set of all possible configuration of $\underline{X}$ on Ω. $\underline{X}$ is MRF with respect to the neighborhood η if a signal has the following Markov property:

$$\Pr(\underline{X} = \underline{x}) > 0 \qquad \text{for all} \quad \underline{x} \in \Gamma \tag{27}$$

$$P(x_s/x_r, r \in \Omega, r \neq s) = P(x_s/x_r, r \in \eta) \tag{28}$$

Compared with Markov chain, one difficulty of MRF is that there is no chain rule for MRF. The joint probability $P(\underline{X} = \underline{x})$ cannot be recursively written in terms of local conditional probability $P(x_s/x_r, r \in \eta)$. Therefore, it is difficult to get the optimal estimation of MRF $\underline{\hat{X}}$, which maximizes the *post-priori* probability:

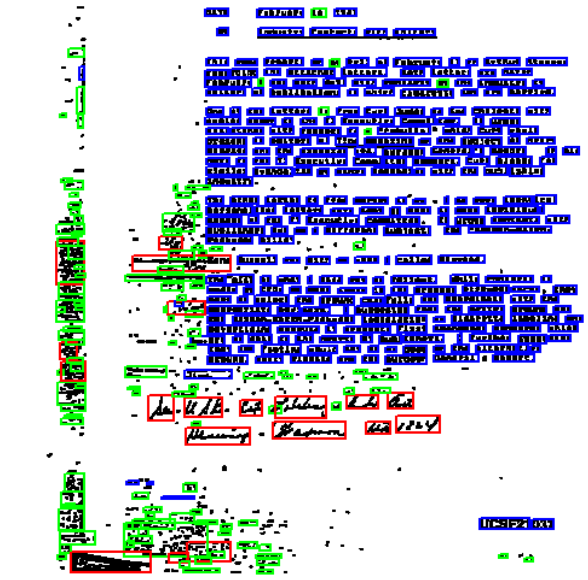
$$\underline{\hat{X}} = arg \max_{\underline{X}} P(\underline{X}/\underline{Y}) \tag{29}$$

The establishment of the connection between MRF and Gibbs distribution provides ways for the optimization of MRF. To maximize the *post-priori* probability of MRF, we need to minimize the total energy of the corresponding Gibbs distribution:

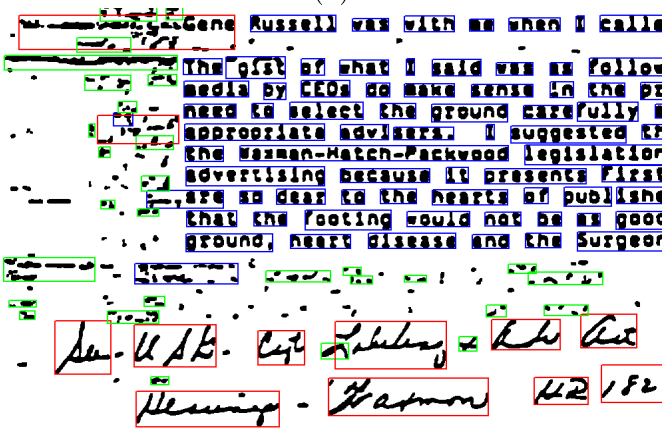
$$\underline{\hat{X}} = arg \min_{\underline{X}} \sum_{c \in \mathcal{C}} V_c(\underline{X}) \tag{30}$$

The value of clique potential $V_c(\underline{X})$ depends on the local configuration on clique c.

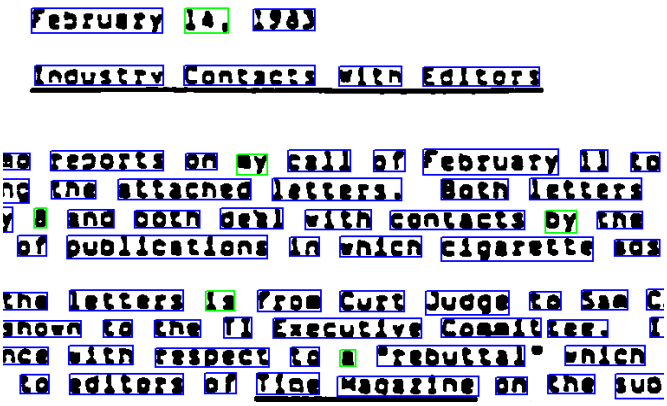
A relaxation algorithm is often used for MRF optimization. There are two types of relaxation algorithms: stochastic and deterministic. Stochastic algorithms can always converge to the global optimal solution if some constraints are satisfied. They are, however, computationally demanding. Deterministic algorithms are lightweight, but only converge to local optimal solution depending on the initial value. In our experiments, Highest Confidence First (HCF), a deterministic approach, is used for MRF optimization due to its fast speed and good performance [9].



(a)



(b)



(c)

Figure 7: Word block segmentation and classification results. Blue, red and green colors represent machine printed text, handwriting and noise respectively. (a) A whole document image, (b) and (c) two parts of the image in (a).

4.2 Clique Definition

As shown in Equation 30, MRF is totally decided by clique c and clique potential $V_c(\underline{X})$. The design of clique and its potential is crucial while the systematical way to design clique is not available yet. In our case, machine printed text, handwriting, and noise exhibit different patterns of geometric relationship. The definition of cliques reflects these differences.

Printed words often form horizontal (or vertical) text lines. Clique C_p is defined in Figure 8(a), which models contextual constraints of neighboring machine printed words. We first define *connection* between word block i and j. As shown in Figure 8(a), O_v is the vertical overlap of two blocks, and D_h is the horizontal distance of two blocks. The distance between block i and j is:

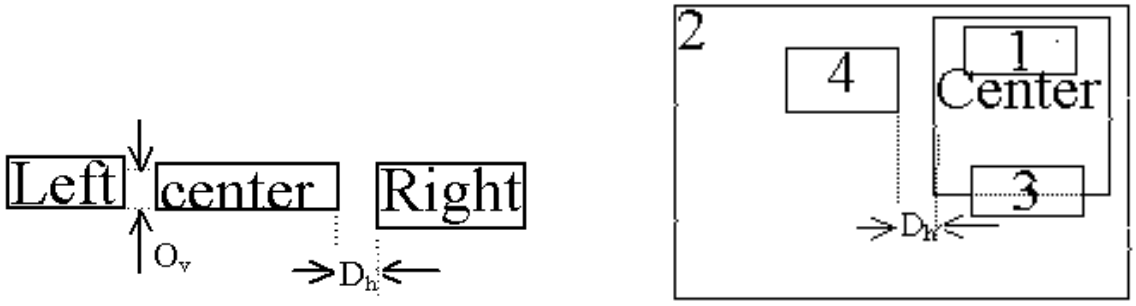
$$D(i,j) = |D_h(i,j) - G_w| + |H_i - H_j| + |Ch_i - Ch_j| \tag{31}$$

where $D_h(i,j)$ is the horizontal distance of word i and j, G_w is the estimated average word gap of the whole document, H_i and H_j are the heights of block i and j respectively, and Ch_i and Ch_j are vertical centers of two blocks. Two blocks are connected if they satisfy:

1. $O_v \geq min(H_i, H_j)/2$
2. $0 \leq D_h \leq 2G_w$
3. $D(i,j) < T_p$, where T_p is a threshold, which is not sensitive to post-processing.

After defining the connection between two blocks, we can construct a graph, in which nodes represent blocks and edges link two connected nodes. The property of an edge can be measured by the distance $D(i,j)$ between two blocks. If a node is connected with more than one node on one side (left or right), we only keep the edge with the smallest distance. Clique C_p can be represented by nodes together with their left and right neighbors.

Noise blocks exhibit rough random pattern in geometric relationship and tend to overlap each other. Figure 8(b) shows clique C_n defined primarily for noise blocks.



(a) (b)

Figure 8: Clique definition. (a) C_p for horizontally arranged machine printed words. (b) C_n for noise blocks.

Similarly, the distance between two blocks is defined as:

$$D(i,j) = \max(D_h(i,j), D_v(i,j)) \tag{32}$$

Where $D_h(i,j) = max(L_i, L_j) - min(R_i, R_j)$, $D_v(i,j) = max(T_i, T_j) - min(B_i, B_j)$, and L, R, T, B are the left, right, top and bottom coordinates of corresponding blocks respectively. If two blocks overlap in horizontal or vertical direction, then $D_h(i,j) < 0$ or $D_v(i,j) < 0$. Two blocks are connected if and only if $D(i,j) < T_n$, where T_n is a threshold. If T_n is too big, wrong label flips of noise and handwriting between two printed text lines may happen. If T_n is too small, the contextual constraints of noise blocks can not be fully used. We set T_n as half of the dominant character height (about 10 pixels in our experiments). Each node, together with all nodes connected with it, are defined as clique C_n. The number of connected nodes may vary from 0 to about 10, depending on the size of the block. As an approximation, we only consider the first 4 nearest connected neighbors. If the number of neighbors is smaller than 4, we set the corresponding neighbors to NULL.

The geometric constraint of handwriting, between machine printed words and noise blocks, has weaker horizontal or vertical structure than machine printed words, and is partially reflected in both clique C_p and C_n. Therefore, we do not define a new specific clique for handwriting.

4.3 Clique Potential

In many applications the clique potentials are defined in ad hoc ways. One systematic way is to define clique potential as the occurrence frequency of each clique in the training set, which can be expressed as a function of local conditional probabilities. The Mobius inversion theorem provides an exact formula for the conversion of local conditional probabilities to clique parameters. It has been shown that if the graph associated with the MRF is cordial, the conversion can be completed exactly; Otherwise the conversion is a good approximation in practice [9]. Based on this idea, we define two clique potentials $V_p(c)$ and $V_n(c)$ for clique C_p and C_n respectively as:

$$V_p(c) = -\frac{P(X_l, X_c, X_r)}{(P(X_l)P(X_c)P(X_r))^w} \tag{33}$$

$$V_n(c) = -\frac{P(x_c, x_1, x_2, x_3, x_4)}{(P(x_c)P(x_1)P(x_2)P(x_3)P(x_4))^w} \tag{34}$$

Where x_l, x_c and x_r are labels for the left, central and right blocks of clique c, w is a constant, and x_i, $i = 1, 2, 3, 4$, is the label of the corresponding ith nearest block. The energy definition for corresponding Gibbs distribution is defined as:

$$U(\underline{X}/\underline{Y}) = -w_s \sum_{s \in \Omega} P(x_s/y_s) + w_p \sum_{c \in C_p} V_p(c) + w_n \sum_{c \in C_n} V_n(c) \tag{35}$$

Where w_s, w_p and w_n are weights, which adjust the relative importance between classification confidence and contextual information of clique C_p and C_n. If $w_s = 1$, $w_p = 0$, and $w_n = 0$, no contextual information is used at all; with the increase of w_p and w_n, more contextual information is emphasized. If we set $w_p = w_n = \infty$, or equivalently set $w_s = 0$, then no classification confidence is used.

In our following experiments, we want to use MRF for word block labeling. The number of handwritten words are much fewer than that of the other two types, leading to smaller estimated frequency of cliques with handwriting. As a result, the optimization tends to label handwritten words as machine printed or noise. Therefore, we regularize the estimated clique frequency $P(x_l, x_c, x_r)$ and $P(x_c, x_1, x_2, x_3, x_4)$ by dividing the products of the probability of each word block label which composes the clique. The above regularization is very similar to the previous approach [57], where w is set to 1. In

our case, w is changeable: increasing w will emphasize handwritten words. Our clique potential definition is very systematic, and can be optimized for different applications.

Figure 9 is an example of the refined classification results after post-processing. Compared with Figure 7, we can see in Figure 9(a) and (b), most mis-classified noise blocks are corrected, with a few exceptions due to less constraints they have. Mis-classified small machine printed words are all corrected in Figure 9(c).

5 Experiments

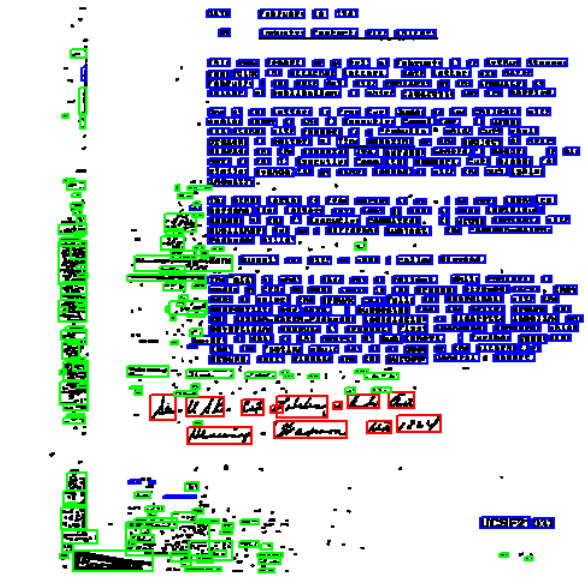
5.1 Classification Accuracy

We first evaluate the classification accuracy without MRF based post-processing. Totally, we collected 318 business letters provided by the tobacco industry. These document images are noisy with lots of handwritten annotations and signatures, few logos, and no figures or tables. At the current stage, we only identify three classes: machine printed text, handwriting and noise. We groundtruthed 94 extremely noisy document images for testing, and keep the rest 224 images for training. All handwritten words (about 1,500) in the training set are groundtruthed. Since there are much more machine printed text and noise blocks, we randomly selected and groundtruthed about the same number of samples of each type in the training set. We use *accuracy* and *precision* as metrics to evaluate the result:

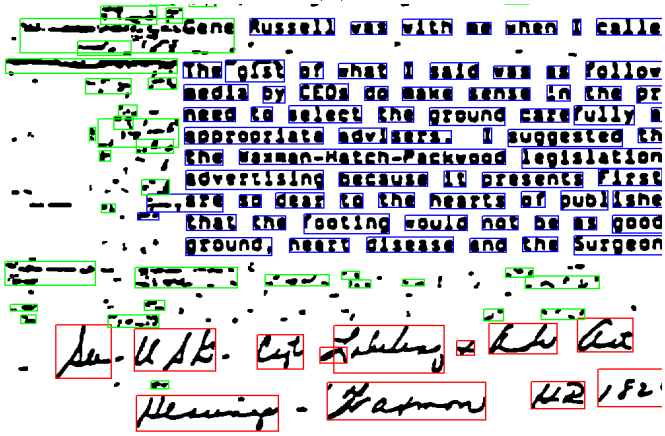
$$\text{Accuracy of type i} = \frac{\text{\# of correctly classified blocks of type i}}{\text{\# of blocks of type i}} \tag{36}$$

$$\text{Precision of type i} = \frac{\text{\# of correctly classified blocks of type i}}{\text{\# of blocks classified as type i}} \tag{37}$$

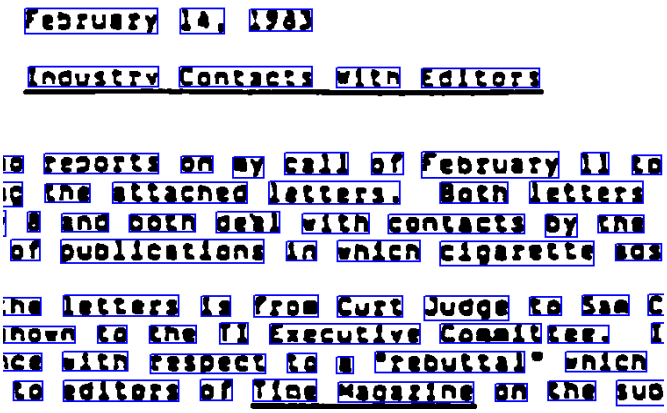
The testing results for 94 images are listed in Table 2. The accuracy of all three classes ranges from 93.2% to 96.8%, with the overall accuracy 96.1%. While the accuracy of handwriting is very high (93.2%), we notice the precision is very low (63.9%). This is mainly because of the small number of handwritten words in the testing set. Even a small percentage of mis-classification from machine printed text and noise to handwriting will significantly decrease the precision of handwriting.



(a)



(b)

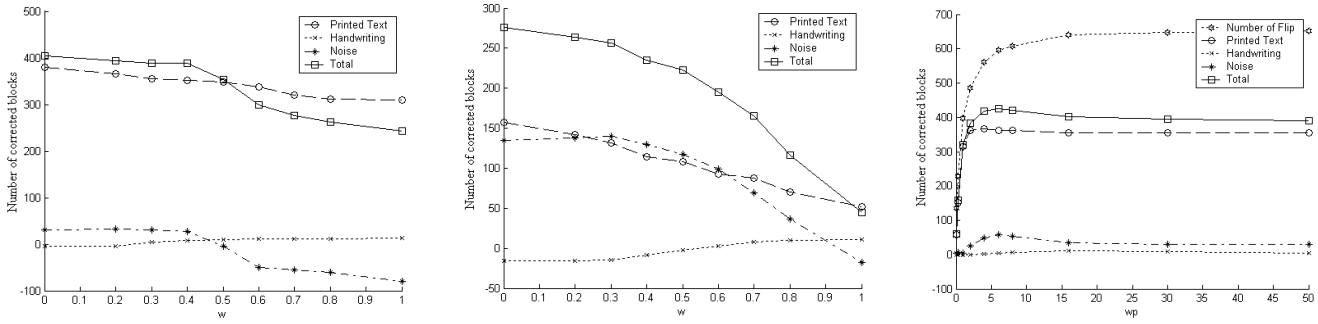


(c)

Figure 9: Word block classification results after post-processing. The results before post-processing is shown in Figure 7. (a) The whole document image, (b) and (c) two parts of the image in (a).

Table 2: Single word block classification

	# of blocks	Percentage	# of correctly classified blocks	# of mis-classified blocks	Accuracy	Precision
Printed text	19,227	66.9%	18,446	781	95.9%	99.5%
Handwriting	701	2.4%	653	48	93.2%	62.9%
Noise	8,802	30.7%	8,522	280	96.8%	93.0%
Total	28,730	100.0%	27,621	1,109	96.1%	N/A



(a) (b) (c)

Figure 10: MRF based post-processing. (a) Number of corrected blocks using clique C_p. (b) Number of corrected blocks using clique C_n. (c) Number of corrected blocks using clique C_p and classification confidence.

5.2 Post-processing Using MRF

In the following experiments we investigate how MRF can improve classification accuracy. In the first run, we set $w_s = 0$, $w_n = 0$ and $w_p = 1$ to show the effectiveness of clique C_p. Figure 10(a) shows the number of corrected blocks, which are previously mis-classified, with the change of w. As expected, C_p is very effective for machine printed words, but not so effective for handwriting and noise. When $w = 0.3$ (under this condition, the classification accuracy of all three classes increases), 355 (46%) of the previously mis-classified machine printed words are corrected. When w increases, handwriting is more emphasized, leading to higher classification accuracy of handwriting, and lower accuracy of machine printed words and noise. In practice, w can be adjusted to optimize the overall accuracy.

In the second run, we test the effectiveness of clique C_n by setting $w_s = 0$, $w_p = 0$

and $w_n = 1$. As shown in Figure 10(b), clique C_n is very effective to correct classification errors of noise blocks. The classification error of noise blocks reduces greatly when w is small. For $w = 0.6$ (under this condition, the classification accuracy of all classes increases), the number of mis-classified noise blocks reduces by 99 (35%). C_n can also correct some classification errors of machine printed words, but less effective than C_p as shown in Figure 10(a).

The third run tests the effectiveness of classification confidence for post-processing. We put classification confidence and clique C_p into one energy function, as shown in Equation 35. Figure 10(c) shows the post-processing result by adjusting w_p when $w = 0.3$, $w_n = 0$ and $w_s = 1$. Adjusting w_p will change the total flip number greatly. When $w_p = 0$, the energy reaches the minimum with the initial labels, and the total flip number is 0. When w_p increases, more emphasis is put on the contextual information, and the flip number increases. When $w_p \to +\infty$, it converges to the case of $w_p = 1$ and $w_s = 0$, the setting of the first run. The maximum of overall classification accuracy is achieved when $w_p = 6$. Compared with the first run, the total number of corrected blocks increases from 389 to 424 by incorporating classification confidence. Similar results are achieved by combining classification confidence with clique C_n.

In the last run, we fix $w_s = 1$ and manually adjust w, w_p and w_n to optimize the overall classification accuracy. The final parameters we choose are $w = 0.39$, $w_p = 5$ and $w_n = 4$. Table 3 shows the results after post-processing. The "Error Reduction Rate" in Table 3 is defined as following:

$$\begin{matrix}\text{Error}\\ \text{Reduction}\\ \text{Rate}\end{matrix} = \frac{\#\text{ of Errors Before Post-Processing } - \#\text{ of Errors After Post-Processing}}{\#\text{ of Error Before Post-Processing}} \tag{38}$$

The mis-classification rate reduces to about half of the original for both machine printed text and noise, but slightly increases for handwriting. However, compared with Table 2, the precision of handwriting increases from 62.9% to 83.3% due to fewer machine printed text and noise mis-classified as handwriting. The overall accuracy increases from 96.1% to 98.1%.

Figure 11 shows another example of machine printed text and handwriting identifi-

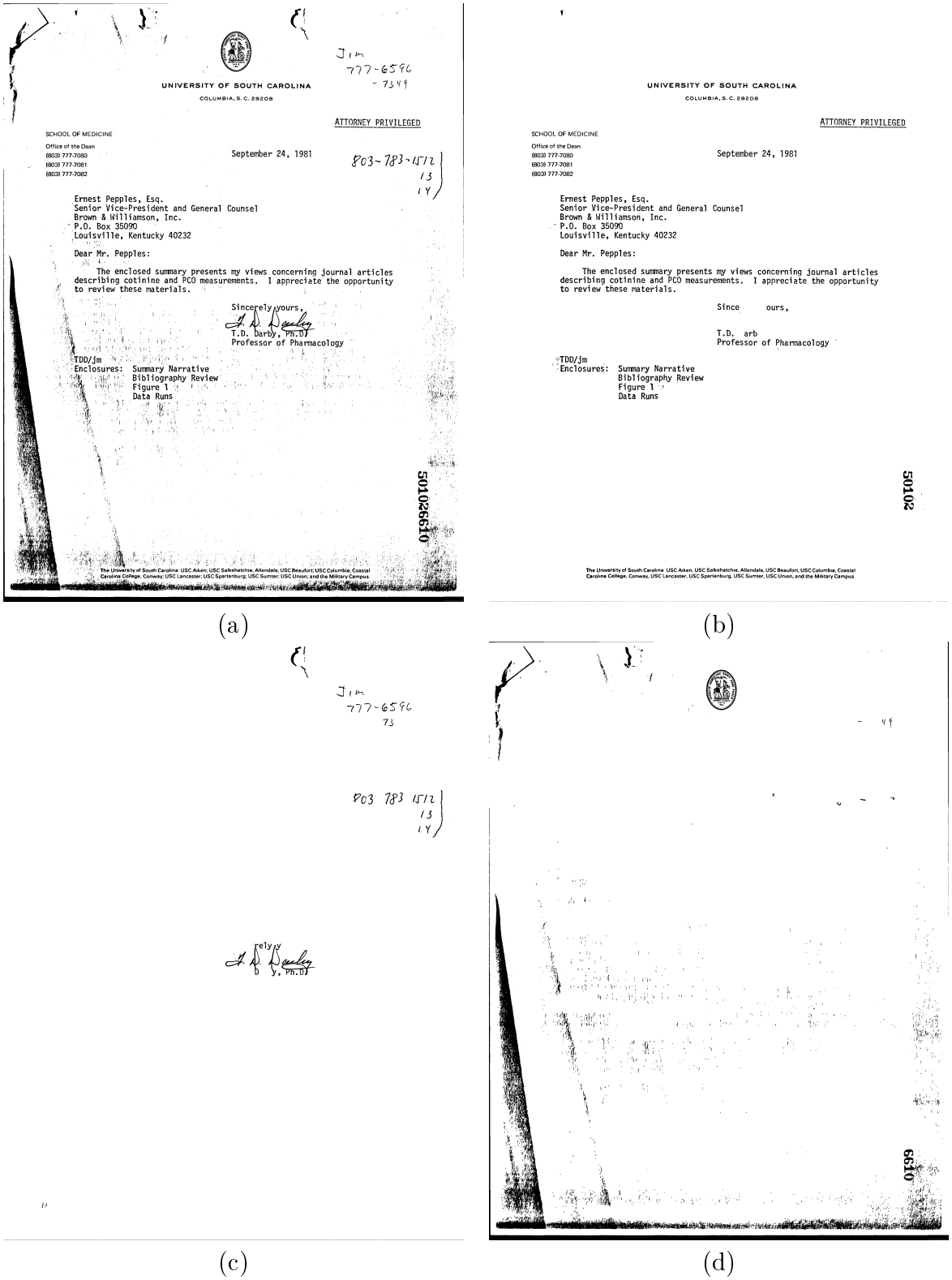


Figure 11: An example of machine printed text and handwriting identification from noisy documents. (a) The original document image, (b) machine printed text, (c) handwriting, (d) noise.

Table 3: Word block classification after MRF based post-processing

	# of blocks	# of correctly classified blocks	# of mis-classified blocks	Reduction of mis-classified blocks	Error reduction rate	Accuracy	Precision
Printed text	19,227	18,835	392	389	49.8%	98.0%	99.7%
Handwriting	701	652	49	-1	-2.1%	93.0%	83.3%
Noise	8,802	8,682	120	160	57.1%	98.6%	96.0%
Total	28,730	28,169	561	548	49.4%	98.1%	N/A

cation from noisy documents. To display the classification results clearly, we decompose the classified image into three layers, representing machine printed text (Figure 11(b)), handwriting (Figure 11(c)), and noise (Figure 11(d)) respectively. The result is very good with few mis-classifications.

Our approach is very general, and can be extended to other languages with minor modification. Figure 12 shows the identification results of a Chinese document. We only need to retrain the classifiers, and the post-processing module is intact. We can see most handwriting and noise blocks are classified correctly, with several machine printed digits are mis-classified as handwriting. On the right margin of the document, some machine printed text is identified as noise due to the touching.

5.3 Page Segmentation in Noisy Images

In this experiment we show our method can improve the general page segmentation results after removing the identified noise. We evaluated two widely used zone segmentation algorithms: Docstrum algorithm [45] and ScanSoft SDK, a commercial OCR software [1]. Many different zone segmentation evaluation metrics have been proposed in previous work. Kanai et al. [26] evaluated zone segmentation accuracy from the OCR respect. Any zone splitting and merging, if not affecting the reading order of text, is not penalized. The approach of Mao et al. is based on text lines, which only penalizes horizontal text line splitting and merging, since it will change the reading order of text [41]. Randriamasy et al. [48] proposed an evaluation method based on multiple ground truth, which is very expensive. Liang's approach is performed at the zone level [38]. He builds the

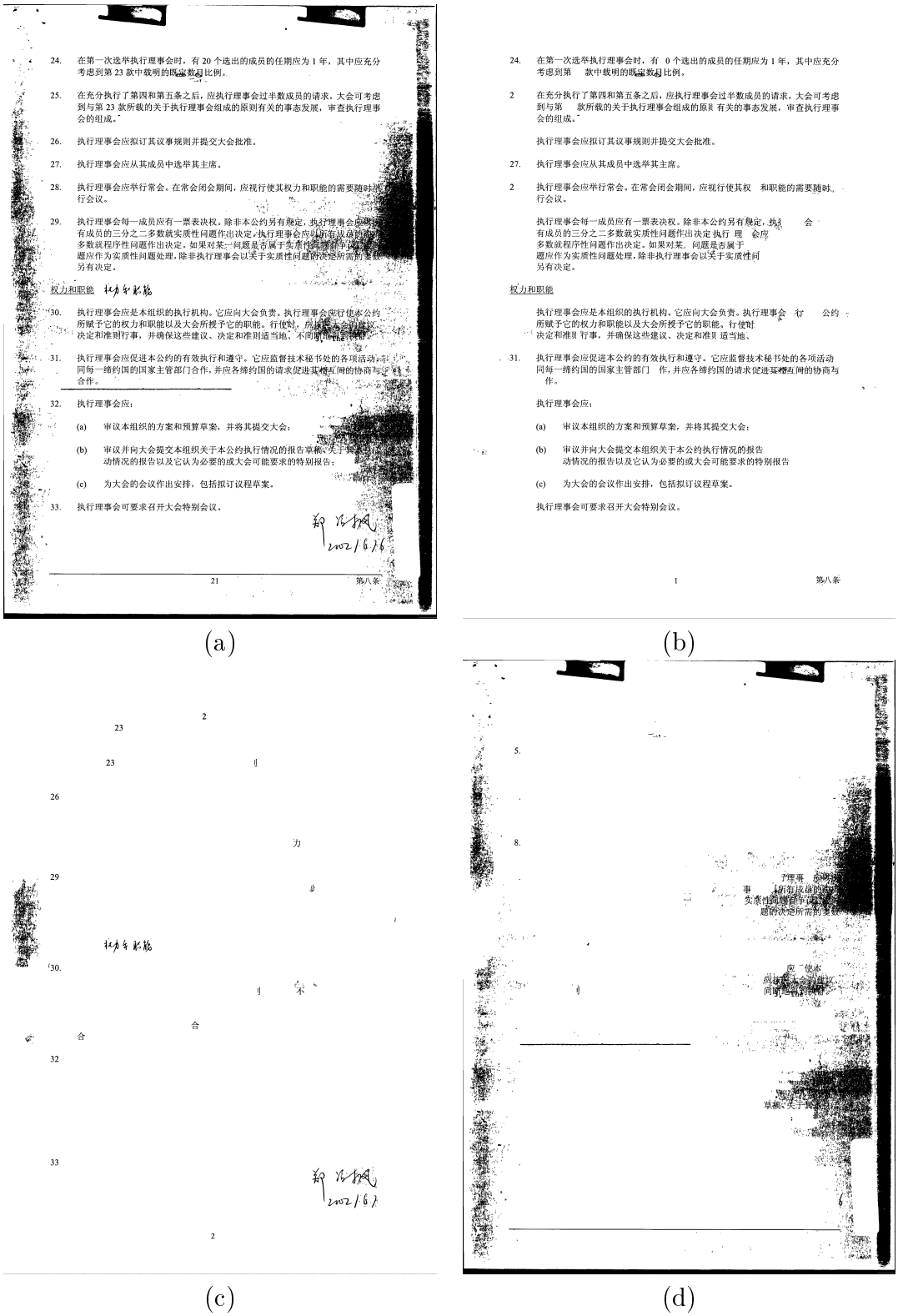


Figure 12: An example of machine printed text and handwriting identification from Chinese documents. (a) Original Chinese document image, (b) machine printed text, (c) handwriting, (d) noise.

correspondence between the segmented and ground truthed zones, and any differences large enough is penalized. We use Liang's scheme in our experiment since we focus more on zone segmentation. From the OCR perspective, vertical splitting or merging of different zones should not be penalized even when these zones have different physical and semantic properties; However, from the point view of zone segmentation, it still should be penalized.

There are 1,374 machine printed text zones in 94 noisy document images. The experiment results are listed in Table 4. All merging and splitting errors are counted as partial correct in the table. Before noise removal, ScanSoft gets very poor results, with the accuracy of 15.9%, on noisy documents under this metrics. After analyzing the segmentation results, we found ScanSoft tends to merge horizontally arrayed zones into one zone, which is suitable for documents with simple layout such as technical articles, but not suitable for other document types such as business letters. Docstrum algorithm outputs much more zones than ScanSoft, resulting in higher accuracy (53.0%), but higher false alarm rate (114.1%) too. After noise removal, the accuracy of both algorithms increases significantly, from 15.9% to 48.4% for ScanSoft and 53.0% to 78.0% for Docstrum algorithm respectively. The false alarm rate reduces from 32.5% to 1.3% for ScanSoft and 114.1% to 7.9% for Docstrum algorithm.

Figure 13 shows the zone segmentation results of two noisy documents with Docstrum algorithm before and after noise removal. The handwriting is outputted to another layer which is not shown here. We can see after noise removal, there are much fewer splitting and merging errors, and overall the segmentation results are significantly improved.

6 Summary

In this paper, we present an approach to identify text from extremely noisy document images. Instead of using some simple filtering rules, we take noise as a distinct class, and use statistical classification techniques to classify each block into machine printed text, handwriting and noise. We then use Markov Random Field to incorporate contextual information for post-processing. Experiments show MRF is a very effective tool to model local dependency among neighboring image components. Our clique potential is defined

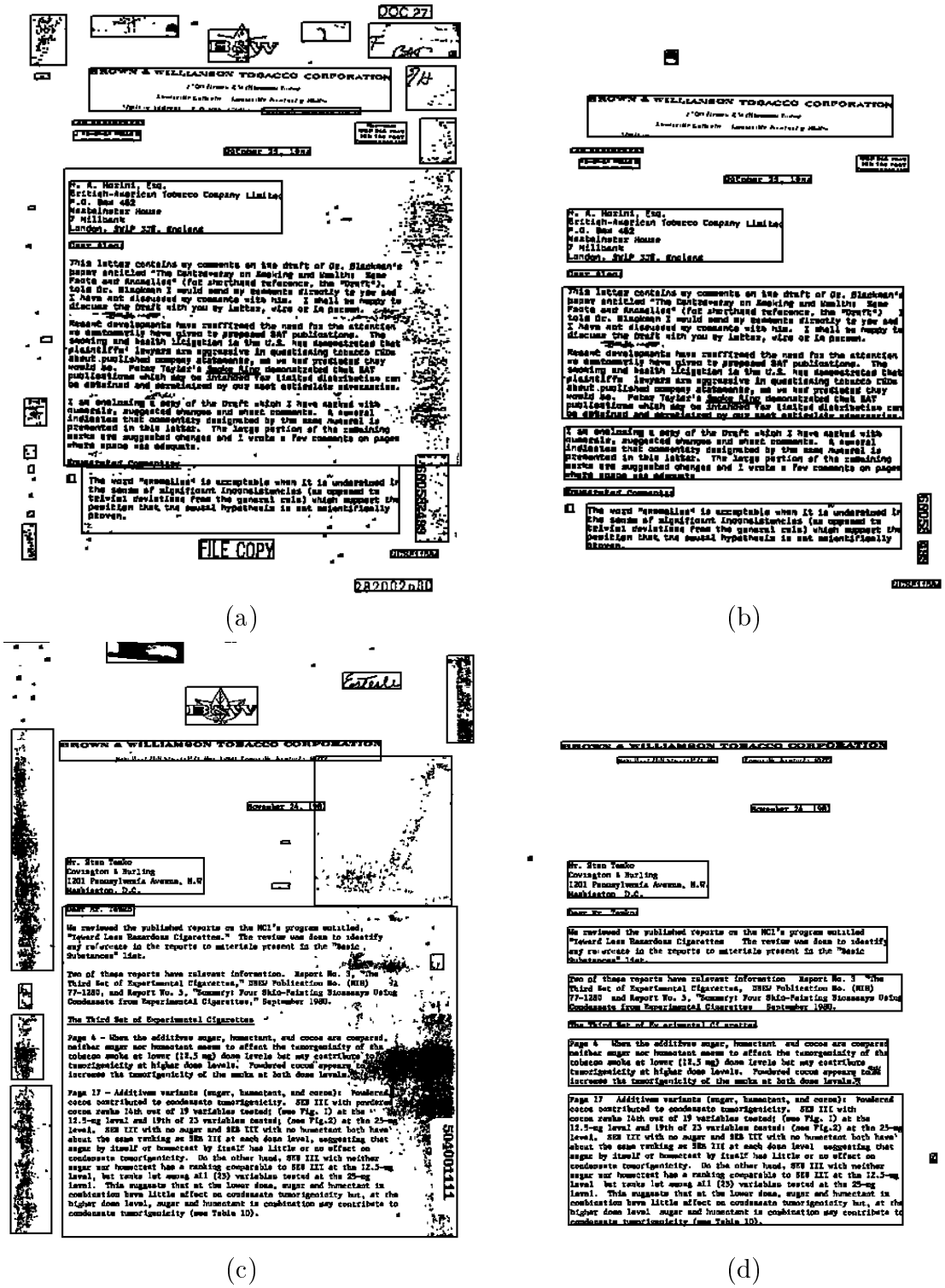


Figure 13: Zone segmentation before and after noise removal using Docstrum algorithm. (a) and (c) show the results before noise removal. (b) and (d) are the results after noise removal.

Table 4: Machine printed zone segmentation experimental results on 94 noisy document images (totally 1,374 zones), before and after noise removal.

	Before noise removal				After noise removal			
	# of correctly segmented zones	# of false alarm zones	# of partially correctly segmented zones	# of missed zones	# of correctly segmented zones	# of false alarm zones	# of partially correctly segmented zones	# of missed zones
ScanSoft	219 (15.9%)	446 (32.5%)	1148 (83.7%)	7 (0.5%)	665 (48.4%)	18 (1.3%)	671 (48.8%)	38 (2.8%)
Docstrum	728 (53.0%)	1568 (114.1%)	646 (47.0%)	0 (0.0%)	1071 (78.0%)	109 (7.9%)	270 (19.7%)	33 (2.4%)

systematically as a function of the joint probability, rather than in an ad hoc way. After post-processing, the classification error rate is reduced to the half of the original. Our method is general enough to be extended to documents of other languages. The technique presented in this paper can be used as an image enhancement to improve page segmentation accuracy of noisy documents. After noise identification and removal, the zone segmentation accuracy increase from 53% to 78% using Docstrum algorithm.

Currently our cliques only use the geometric contextual relationship within neighboring word blocks. Each clique with the same type has the same potential which may loss useful information. For example, for clique C_p, the clique of three printed words with roughly the same height is quite different from those with different heights. In the latter case, it is possible that one of the blocks is erroneously identified. Another potential improvement is to integrate high level contextual information besides the local contextual information we used. For example, the text line and zone segmentation results can be fed back to our classification module to refine the classification. To effectively use contextual information is one of our future research directions.

References

[1] ScanSoft developer's kit 2000. ScanSoft Corp., http://www.scansoft.com, 2000.

[2] T. Akiyama and N. Hagita. Automated entry system for printed documents. *Pattern Recognition*, 23(11):1141–1154, 1990.

[3] M.B.H. Ali. Background noise detection and cleaning in document images. In *Proceedings of the* 13^{th} *International Conference on Pattern Recognition*, pages 758–762, 1996.

[4] H.S. Baird. Calibration of document image defect models. In *Proceedings* 2^{nd} *Annual Symposium on Document Analysis and Information Retrieval*, pages 1–16, 1993.

[5] H.S. Baird. Document image quality: Making fine discriminations. In *Proceedings of the* 5^{th} *International Conference on Document Analysis and Recognition*, pages 459–462, 1999.

[6] H.S. Baird, S.E. Jones, and S.J. Fortune. Image segmentation by shape-directed covers. In *Proceedings of the* 10^{th} *International Conference on Pattern Recognition*, pages 820–825, 1990.

[7] M. Cannon, J. Hochberg, and P. Kelly. Quality assessment and restoration of typewritten document images. *International Journal on Document Analysis and Recognition*, 2:80–89, 1999.

[8] K. Chinnasarn, Y. Rangsanseri, and P. Thitimajshima. Removing salt-and-pepper noise in text/graphics images. In *The 1998 IEEE Asia-Pacific Conference on Circuits and Systems*, pages 459–462, 1998.

[9] P.B. Chou, P.R. Cooper, and M.J. Swain. Probabilistic network inference for cooperative high and low level vision. In R. Chellapa and A.K. Jain, editors, *Markov Random Fields: Theory and Application*. Academic Press Inc, 1993.

[10] D. Doermann and J. Liang. Binary document image using similarity multiple texture features. In *Proceedings of The 2001 Symposium on Document Image Understanding Technology*, pages 181–193, 2001.

[11] K. Etemad, D. Doermann, and R. Chellappa. Multiscale document page segmentation using soft decision integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):92–96, 1997.

[12] K.-C. Fan, Y.-K. Wang, and T.-R. Lay. Marginal noise removal of document images. In *Proceedings of the* 6^{th} *International Conference on Document Analysis and Recognition*, pages 317–321, 2001.

[13] K.C. Fan, L.S. Wang, and Y.T. Tu. Classification of machine-printed and handwritten texts using character block layout variance. *Pattern Recognition*, 31(9):1275–1284, 1998.

[14] J. Fanke and M. Oberlander. Writing style detection by statistical combination of classifier in form reader applications. In *Proceedings of the* 2^{nd} *International Conference on Document Analysis and Recognition*, pages 581–585, 1993.

[15] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, 2^{nd} edition, 1990.

[16] D. Gabor. Theory of communication. *J. Inst. Elect. Engr*, 93:429–459, 1946.

[17] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[18] G. Grimmett and D. Stirzaker. *Probability and Random Processes.* Oxford University Press, 2^{nd} edition, 2001.

[19] J.K. Guo and M.Y. Ma. Separating handwritten material from machine printed text using hidden Markov models. In *Proceedings of the* 6^{th} *International Conference on Document Analysis and Recognition*, pages 439–443, 2001.

[20] R.M. Haralick. Document image understanding: Geometric and logical layout. In *Proceedings of Computer Vision and Pattern Recognition*, pages 385–390, 1994.

[21] J.D. Hobby and T.K. Ho. Enhancing degraded document images via bitmap clustering and averaging. In *Proceedings of the* 4^{th} *International Conference on Document Analysis and Recognition*, pages 394–400, 1997.

[22] J.J. Hull. Incorporating language syntax in visual text recognition with a statistical model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1251–1256, 1996.

[23] A.K. Jain and S. Bhattacharjee. Text segmentation using Gabor filters for automatic document processing. *Machine Vision and Applications*, 5:169–184, 1992.

[24] A.K. Jain and B. Yu. Document representation and its application to page decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):294–308, 1998.

[25] A.K. Jain and D. Zongker. Feature selection: Evaluation, application and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[26] J. Kanai, S.V. Rice, T.A. Nartker, and G. Nagy. Automated evaluation of OCR zoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):86–90, 1995.

[27] T. Kanungo, H.S. Baird, and R.M. Haralick. Validation and estimation of document degradation models. In *Proceeding of* 4^{th} *Annual Symposium on Document Analysis and Information Retrieval*, pages 217–228, 1995.

[28] T. Kanungo, R. M. Haralick, and I. Phillips. Nonlinear local and global document degradation models. *International Journal of Imaging Systems and Technology*, 5(4):220–230, 1994.

[29] T. Kanungo, R.M. Haralick, H.S. Baird, W. Stuetzle, and D. Madigan. Document degradation models: Parameter estimation and model validation. In *Proceeding of International Workshop on Machine Vision Applications*, pages 552–557, 1994.

[30] T. Kanungo and Q. Zheng. Estimation of morphological degradation model parameters. In *Proceedings of IEEE International Conference on Speech and Signal Processing*, pages 1961–1964, 2001.

[31] K. Kuhnke, L. Simoncini, and Zs. M. Kovacs-V. A system for machine-written and hand-written character distinction. In *Proceedings of the* 3^{rd} *International Conference on Document Analysis and Recognition*, pages 811–814, 1995.

[32] L. Kukolick and R. Lippmann. LNKnet user's guide. MIT Lincoln Laboratory, http://www.ll.mit.edu/IST/lnknet/, May 1999.

[33] S.-W. Lee and B.-S. Ryu. Parameter-free geometric document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1240–1256, 2001.

[34] H. Li and D.S. Doermann. Text quality estimation in video. In *Proceedings of the SPIE Vol. 4670 - Document Recognition & Retrieval IX*, pages 232–243, 2002.

[35] H. Li, O. Kia, and D. Doermann. Text enhancement in digital video. In *Proceedings of the SPIE Vol. 3027 - Document Recognition IV*, pages 1–8, 1999.

[36] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2^{nd} edition, 2001.

[37] J. Liang and R.M. Haralick. Document image restoration using binary morphological filters. In *Proceedings of the SPIE Vol. 2660 - Document Recognition III*, pages 274–285, 1996.

[38] J. Liang, I.T. Phillips, and R.M. Haralick. Performance evaluation of document layout analysis algorithms on the UW data set. In *Proceedings of the SPIE Vol. 3027 - Document Recognition IV*, pages 149–160, 1997.

[39] X. Lin, X. Ding, and M. Chen. Adaptive confidence transform based classifier combination for Chinese character recognition. *Pattern Recognition Letters*, 19(10):975–988, 1998.

[40] R.P. Loce and E.R. Dougherty. *Enhancement and Restoration of Digital Documents – Statistical Design of Nonlinear Algorithms*. SPIE Optical Engineering Press, 1997.

[41] S. Mao and T. Kanungo. Empirical performance evaluation methodology and its application to page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):242–256, 2001.

[42] G. Nagy, S. Seth, and S. Stoddard. Document analysis with an expert system. In *Pattern Recognition in Practice II*, pages 149–155. Elsevier Science, 1984.

[43] H. Nishida and T. Suzuki. Correcting show-through effects on document images by multiscale analysis. In *Proceedings of the* 16^{th} *International Conference on Pattern Recognition*, 2002.

[44] L. O'Gorman. Image and document processing techniques for the RightPages electronic library system. In *Proceedings of the* 11^{th} *International Conference on Pattern Recognition*, pages 820–825, 1992.

[45] L. O'Gorman. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173, 1993.

[46] V. Pal and B.B. Chaudhuri. Machine-printed and handwritten text lines identification. *Pattern Recognition Letters*, 22(3-4):431–441, 2001.

[47] T. Pavlidis and J. Zhou. Page segmentation and classification. *CVGIP: Graphical Models and Image Processing*, 54(6):484–496, 1992.

[48] S. Randriamasy, L. Vincent, and B. Wittner. An automatic benchmarking scheme for page segmentation. In *Proceedings of the SPIE Vol. 2181 - Document Recognition*, pages 217–227, 1994.

[49] R.M.K. Sinha, B. Prasada, G.F. Houles, and M. Sabourin. Hybrid contextual text recognition with string matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):915–925, 1993.

[50] A. Soffer. Image categorization using texture features. In *Proceedings of the* 4^{th} *International Conference on Document Analysis and Recognition*, pages 233–237, 1997.

[51] S.N. Srihari, Y.C. Shim, and V. Ramanprasad. A system to read names and address on tax forms. Technical Report CEDAR-TR-94-2, CEDAR, SUNY, Buffalo, 1994.

[52] P. Stubberud, J. Kanai, and V. Kalluri. Adaptive image restoration of text images that contain touching or broken characters. In *Proceedings of the* 3^{rd} *International Conference on Document Analysis and Recognition*, pages 778–781, 1995.

[53] S. Sural and P.K. Das. A two-state Markov chain model of degraded document images. In *Proceedings of the* 5^{th} *International Conference on Document Analysis and Recognition*, pages 463–466, 1999.

[54] D. Sylwester and S. Seth. Adaptive segmentation of document images. In *Proceedings of the* 6^{th} *International Conference on Document Analysis and Recognition*, pages 827–831, 2001.

[55] Q. Wang and C.L. Tan. Matching of double-sided document images to remove interference. In *Proceedings of Computer Vision and Pattern Recognition*, 2001.

[56] Y. Wang, R.M. Haralick, and I.T. Phillips. Zone content classification and its performance evaluation. In *Proceedings of the* 6^{th} *International Conference on Document Analysis and Recognition*, pages 540–544, 2001.

[57] C. Wolf and D. Doermann. Binarization of low quality text using a Markov random field model. In *Proceedings of the* 16^{th} *International Conference on Pattern Recognition*, 2002.

[58] Y. Zhang and R.R. Loce. Document restoration and enhancement using optimal iterative and paired morphological filters. In *Proceedings of the SPIE Vol. 3027 - Document Recognition IV*, pages 109–123, 1997.

[59] Z. Zhang and C.L. Tan. Recovery of distorted document images from bound volumes. In *Proceedings of the* 6^{th} *International Conference on Document Analysis and Recognition*, pages 429–433, 2001.

[60] Y. Zheng, H. Li, and D. Doermann. The segmentation and identification of handwriting in noisy document images. In *Proceedings of the* 5^{th} *International Workshop on Document Analysis Systems*, pages 95–105, 2002.

[61] Y. Zheng, C. Liu, and X. Ding. Single character type identification. In *Proceedings of the SPIE Vol. 4670 - Document Recognition & Retrieval IX*, pages 49–56, 2002.