2011 International Conference on Document Analysis and Recognition

Fast Rule-line Removal using Integral Images and Support Vector Machines

Jayant Kumar David Doermann
Institute of Advanced Computer Studies
University of Maryland College Park, USA
{*jayant,doermann*}*@umiacs.umd.edu*

***Abstract*—In this paper, we present a fast and effective method for removing pre-printed rule-lines in handwritten document images. We use an integral-image representation which allows fast computation of features and apply techniques for large scale Support Vector learning using a data selection strategy to sample a small subset of training data. Results on both constructed and real-world data sets show that the method is effective for rule-line removal. We compare our method to a subspace-based method and show that better accuracy can be achieved in considerably less time. The integral-image based features proposed in the paper are generic and can be applied to other problems as well.**

***Keywords*-Rule-line, Handwritten Documents, Arabic**

I. Introduction

In handwritten documents, it is very common to find rule-lines used to keep the base line of written content straight. But when it comes to processing the images, the lines can make document analysis much more difficult. For example, horizontal rule-lines tend to "connect" the characters/words making connected-component methods unreliable. If the page has vertical lines for its margin then all the text lines may become connected making the whole foreground content a single connected-component. Text line extraction [3], page segmentation and character recognition methods which are based on the assumption that characters and/or words form separate connected-components fail to work in these scenarios. Hence it becomes extremely important to remove rule-lines in the pre-processing step with the minimal effect on the quality of text.

Rule-line removal is still considered a difficult task due to the fluctuation in thickness of rule-lines and the large variation in shape of handwritten characters. Existing approaches often fail when the lines are severely broken, not straight, and/or when the rule-lines interact significantly with the text [5]. As shown in Figure 1, different parts of the same rule-line exhibit different characteristics with respect to level of degradation, thickness and radius of curvature. These variations are introduced during the collection, scanning and binarization of the documents. Features extracted from the rule-lines and the regular baselines of Arabic characters have very similar distributions which may lead to ambiguity in segmentation and recognition of words and characters. Figure 2 shows some example scenarios where the base-line of Arabic characters completely overlap the rule-line. It is

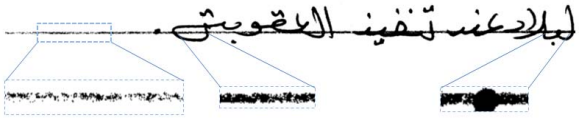

Figure 1. Three different parts of the same rule-line having different width and level of degradation.

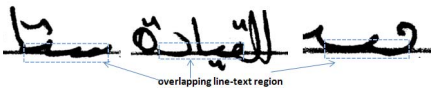

Figure 2. Example of scenarios where Arabic characters exhibit long baseline. The overlap with similar width rule-line makes the separation very difficult.

even difficult for humans, to mark the exact boundaries of baselines in Arabic scripts [5].

Existing approaches for rule-line removal can be broadly classified as *heuristic-based* or *model-based*. In *heuristic-based* approaches, rule-lines are detected and removed using the *Projection profiles*, *Hough-transform*, *Run-lengths* or *Morphological operators*. For *Projection profile* based methods, a horizontal histogram is used to locate the center locations of horizontal rule-lines [1], [5]. These methods are very sensitive to the skew of the document image and often have difficulty in estimating the accurate thickness of line. To overcome these problems, Cao *et al.* partitioned the image into vertical zones and projection profiles were computed for each zone [5]. But finding the optimal width of vertical zone is difficult and empirical. *Hough-transform* based methods detect lines based on peaks in parameter-space also known as Hough-space. They can detect broken-lines but may fail to accurately localize rule-lines with varying thickness. To address this problem, Chen and Lee proposed the *strip projection* method motivated by the fact that lines are more likely to form peaks in a small region [7]. But one of the main drawbacks of *Hough-based* methods are that they are computationally slow. *Morphological operator* based methods use a structuring element to remove rule-lines by dilation and erosion operations [8]. The design of accurate structuring elements often depends on the width of the rule-line and the strokes of characters. These methods are incapable of removing rule-lines with large variation in

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.123

IEEE computer society

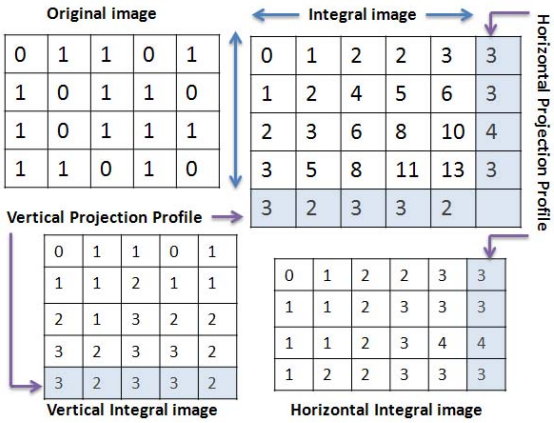

Figure 3. The value at any point (x,y) of integral image is the sum of all pixel values above and to the left. If we compute horizontal and vertical integral image then any row and column sum can be computed in two array references.

thickness. Shi *et al.* used directional local profiling followed by adaptive vertical run-length search to remove rule-lines in Arabic documents [6]. Although the method seems to work reasonably well it is susceptible to remove text pixels in overlapping regions, thereby degrading the quality of text. Abd-Almageed *et al.* proposed a *linear-subspace* based method to detect rule-line pixels in binary images [2]. The computation of central-moment features used in their approach is very time-consuming for all the foreground pixels and makes the method infeasible for high-resolution document images.

Although at a very basic level, text/rule-line separation is a classification problem, until recently researchers have avoided modeling the problem in this way. The main bottleneck of taking a pixel-based classification approach for rule-line removal is the feature computation and classification time for each pixel. Traditional feature extraction and supervised learning approaches can be very time-consuming. Support Vector machine (SVM), which is one of the most popular method for supervised classification, due to its quadratic time ($O(n^2)$) dependency on the size of data, cannot handle large training sets. To train 10 million 60-dimensional points, it takes 24 hours with just one set of parameters. In this scenario, obtaining an optimal set of parameters using a grid search may take several months. In the formulation of SVM, however, the final decision surface depends on only small subset of training data called Support vectors (SVs) [11]. Hence for large datasets it becomes important to first select points which are likely to be SVs and then solve a much smaller quadratic programming problem.

In this paper we first propose features based on an integral-image representation [9] which are not only discriminative for text/rule-line classification but are also very fast to compute. Once the integral-image is computed, feature

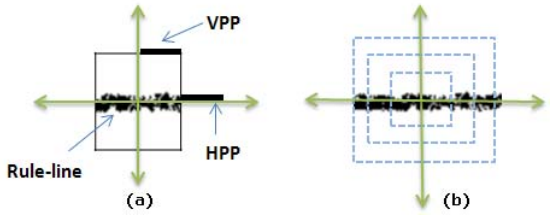

Figure 4. Illustration of features computation. (a) HPP and VPP in the first quadrant. Features from all the four quadrants are concatenated to form a feature vector at one scale. (b) Three different scales for computing features

computation for each pixel is just few subtraction operations. We express the computation of *Horizontal Projection Profile* (HPP) and *Vertical Projection profile* (VPP) around each pixel in terms of integral image, thereby making the computation very fast. Second, we present and employ a method to select probable support-vector points from a large training set to do fast SVM training [13]. We test our rule-line removal algorithm using constructed and real-world handwritten Arabic document images which contain pre-printed horizontal and vertical rule-lines. Our experiments show that the proposed approach is effective and computationally feasible even for high-resolution document images. The idea proposed in this paper is very generic and can be applied to any problem where pixel-level feature computations and large scale SVM training are required.

The remainder of the paper is organized as follows. In Section II we present the proposed rule-line removal approach and provide details of evaluation and datasets used in our experiments in Section III. We present our experimental results in Section IV and conclude our paper in Section V.

II. Rule-line Removal

A. Integral Image Features

HPP and VPP can be computed very rapidly using an intermediate representation of image known as an integral-image [9]. The value of integral image (ii) at the location (x,y) is the sum of pixel values above and to the left of (x,y) as demonstrated in Figure 3.

$$ii(x,y) = \sum_{x\prime \le x, y\prime \le y} i(x\prime, y\prime) \quad (1)$$

An integral-image can be computed in one pass over the original image. Using the integral image any rectangular sum can be computed in four array references. Moreover, any row sum and column sum can be computed in just two array references if we compute both the vertical (V_{ii}) and horizontal (H_{ii}) integral image (Figure 3).

$$H_{ii}(x,y) = \sum_{x\prime \le x} i(x\prime, y), \quad V_{ii}(x,y) = \sum_{y\prime \le y} i(x, y\prime) \quad (2)$$

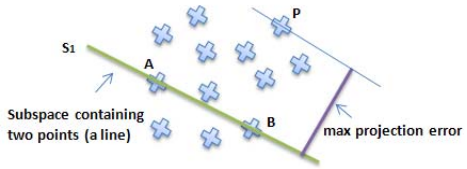

Figure 5. S_1 is a subspace containing points A and B. The point with maximum projection error (P) lies on the Convex-hull of points.

Figure 6. Rule-line removal is done in two steps. Two different classifiers are used to remove the rule-line only region and the mixed region in different steps.

$$HPP_k(c_1, c_2) = H_{ii}(k, c_2) - H_{ii}(k, c_1) \tag{3}$$

$$VPP_k(r_1, r_2) = V_{ii}(k, r_2) - V_{ii}(k, r_1) \tag{4}$$

where $HPP_k(c_1, c_2)$ represents the sum of k^{th} row between columns c_1 and c_2 and $VPP_k(r_1, r_2)$ represents the sum of k^{th} column between rows r_1 and r_2. As observed in Equation 3 and Equation 4 the computation of each sum takes one subtraction and two array accesses.

We use the HPP and the VPP of the four quadrants around the pixel as features to train a two-class SVM. We concatenate these features at different scales to capture more context around each pixel. Figure 4(a) shows an illustrative example for HPP and VPP in the first quadrant of a rule-line image. Figure 4(b) shows the different scales considered for feature computation. If we consider a quadrant of size 16x16 then the computation of HPP and VPP requires 64 = (16+16)*2 array accesses instead of 256 = 16*16 accesses without integral-image.

B. Data Selection for Support Vector Learning

Since the number of foreground pixels in a high-resolution document image may be large we also apply a data selection method to select a small subset of training data. Our selection technique is motivated by the fact that SVs are subset of the points in Convex-hulls of data sets (in linearly separable case). Our method is based on *incremental subspace learning*, which has been extensively used in pattern recognition and computer vision [12]. The idea is that one starts with a subspace of dimension zero ($\mathbf{S_0}$) with a single point ($\mathbf{v_k}$) and incrementally adds points from the training set using the following strategy.

$$\mathbf{S_0} = \left[\frac{\mathbf{v_k}}{\| \mathbf{v_k} \|}\right] \tag{5}$$

If the newly selected point lies in the current subspace, then we do not do anything and find another point. If it does not, we create a new subspace which has dimension one higher than the previous subspace ($\mathbf{S_{d+1}}$), to accommodate the new point. For this we first compute the projection $\mathbf{p}$ of the new vector $\mathbf{v_i}$ on current subspace $\mathbf{S_d}$:

$$\mathbf{p} \leftarrow \mathbf{S_d}^T \mathbf{v_i} \tag{6}$$

where the superscript (T) represents to the transpose of matrix $\mathbf{S_d}$. In the next step we compute the reconstructed ($\mathbf{r_v}$) and the residual vector ($\mathbf{v_{res}}$) as follows :

$$\mathbf{r_v} \leftarrow \mathbf{S_d p} \tag{7}$$

$$\mathbf{v_{res}} \leftarrow \mathbf{r_v} - \mathbf{v_i} \tag{8}$$

$$\mathbf{S_{d+1}} = \left[\mathbf{S_d} \quad \frac{\mathbf{v_{res}}}{\| \mathbf{v_{res}} \|}\right] \tag{9}$$

Finally, we obtain a subspace containing all the points. Our method uses this strategy to select points, since for every new subspace, the point not contained in the subspace and having the maximum projection error (ϵ_i) will lie on the Convex-hull of training points of each class (Figure 5).

$$\epsilon_i = d(\mathbf{r_v}, \mathbf{v_i}) = \sqrt{\sum |r_v - v_i|^2} \tag{10}$$

$$\mathbf{v_p} = \{\mathbf{v_i} : \epsilon_i > \epsilon_k \quad \forall k\} \tag{11}$$

where $d(.,.)$ represents the Euclidian distance between two vectors. $\mathbf{v_p}$ is the point with maximum projection error. For the non-linearly separable case we replace the dot-products by kernel function and use slack-variables to sample points in feature space. More details on this approach can be found in [13].

C. SVM based Text-Ruleline Classifier

We train a C-SVM [14] classifier using the selected training points. The classification time for SVM depends on the number of SVs in the trained model. As more and more points lie near the boundary of separation of two-classes, the number of SVs also increases. To expedite the rule-line removal we train two SVM classifiers. The first classifier is trained to remove rule-line pixels from *rule-line only* regions as shown in Figure 6. We only use the features extracted at

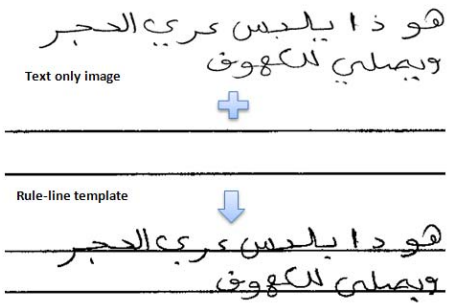

Figure 7. Creation of constructed data from a text-only image and a rule-line template.

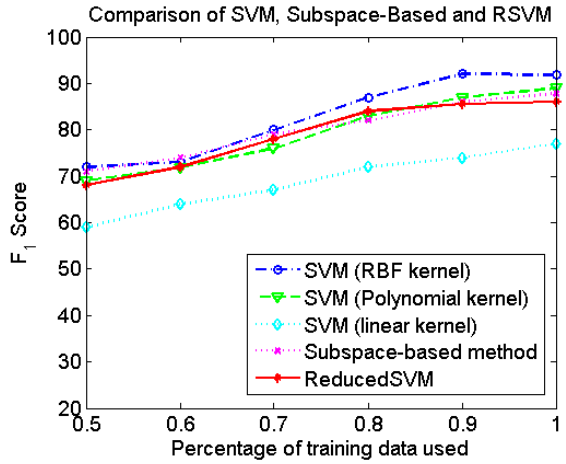

Figure 8. Plot of F1-scores on constructed data

one scale (4x4 quadrants) to train this classifier. The second classifier is used in second step to remove rule-lines from the *mixed-region* (Figure 6). In this stage we compute features from three windows at different scales (4x4, 6x6, 8x8 quadrants). This is computationally more efficient because the first classifier is less computationally intense with much fewer SVs. Another advantage of detecting the rule-line pixels in first pass is that using those pixels we can estimate the parameters of each rule-line and use it to restrict the number of pixels to be classified in second pass.

III. Datasets and Evaluation

Evaluating methods for pixel-level content separation requires pixel-level annotation of images. Such annotation is difficult to obtain manually for large collections of document images. Due to this we evaluate our approach using a constructed dataset proposed in [2] and a small set of real handwritten document images. Images in the first dataset were created by combining templates of rule-line images with the images containing handwritten text (Figure 7). It contains a total of 50 images each having a resolution of 300 dpi. We also report results on a dataset of 10 real

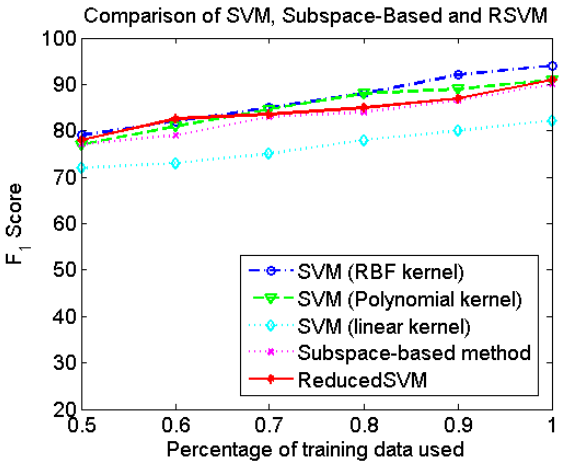

Figure 9. Plot of F1-scores on real-data

handwritten images with pre-printed rule-lines. The ground-truth for these images were created using $Microsoft^{\circledR}$ *Paint* by removing rule-line pixels manually. The resolution of each image is 600 dpi. Both the datasets are available for download at [15].

We compute *recall* and *precision* values along with their harmonic mean (F_1 score) to evaluate our method. If a rule-line pixel detected by our method is also a rule-line pixel in ground-truth then it is counted as *true positive* (TP). Similarly, if a rule-line pixel detected by our method is not a rule-line pixel in ground-truth then it contributes to *false positive* (FP). *False negatives* (FN) are those rule-line pixels which are missed by our algorithm. Using these values we compute *precision*, *recall* and F_1 score as follows:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F_1 score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{13}$$

IV. Experimental Results

In our first set of experiments we divided the dataset used in [2] into training and test images. A total of 35 training images were used to obtain the plots shown in Figure 8. An SVM with a radial-basis kernel (RBF) achieves the best accuracy of 91.4% on test images. The poor accuracy of the linear-kernel confirms that the two classes are not linearly separable. We also evaluated our method on 10 real handwritten images. A similar plot of F_1 with increasing percentage of training data is shown in Figure 9. We used 70% of the available data for training. Slightly better accuracy (94%) for second dataset may be due to the fact that in real scenarios, the possible interactions between rule-lines and text is more structured and limited than random

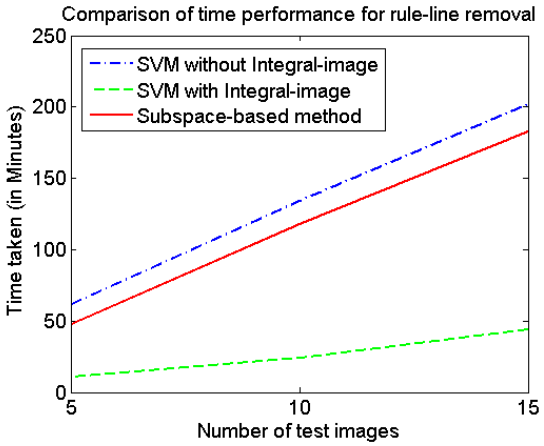

Figure 10. Comparison of time-performance on test data

interactions in the constructed data. In order to compare our data selection strategy for large scale SVM learning, we used a Reduced-SVM [10] for training and obtained a similar plot of F1-scores on test data. Reduced-SVM uses a randomly selected subset of the data (typically less than 10% of original data) to obtain a nonlinear separating surface for classification. Misclassified pixels in our results are mainly from the mixed-regions where a clear-cut boundary between the rule-line and text is ambiguous. In Figure 10, we show the plots of time-taken for rule-line removal. As observed, there is a significant reduction in time using the integral-image features. All the experiments were conducted on a P4 machine with 3GB RAM.

V. Conclusion

In this paper we first presented a fast way to compute *Horizontal Projection Profile* and *Vertical Projection Profile* features using integral images. We then applied a data selection method for large scale SVM training. We showed in our experiments that the integral-image features are effective for text/rule-line classification. We then showed that the computation time for rule-line removal reduces significantly if we use integral-image. We experimented with high resolution images (300 dpi and 600 dpi) to demonstrate that using large scale learning techniques, pixel-based rule-line removal is feasible. In the future we plan to demonstrate additional problems where integral-image based features can be applied to reduce the computation time.

Acknowledgment

The partial support of this research by DARPA through BBN/DARPA Award HR0011-08-C-0004 under subcontract 9500009235, the US Government through NSF Award IIS-0812111 is gratefully acknowledged.

References

[1] K. R. Arvind, J. Kumar, A. G. Ramakrishnan, *Line removal and Restoration of Handwritten strokes*, Intl. Conf. on Comp. Intelligence and Multimedia Applications, IEEE CS Press Volume:3, pp. 208–214, 2007

[2] W. Abd-Almageed, J. Kumar and D. Doermann. *Page Rule-Line Removal using Linear Subspaces in Monochromatic Handwritten Arabic Documents*, Intl. Conf. on Document Analysis and Recognition, pp. 768–772, 2009.

[3] J. Kumar, W. Abd-Almageed, L. Kang and D. Doermann, *Handwritten Arabic Text Line Segmentation using Affinity Propagation*, Document Analysis Systems, pp. 135–142, 2010.

[4] A. K. Chhabra, V. Misra, and J. F. Arias. *Detection of horizontal lines in noisy run length encoded images: The fast method*, Intl. Work. on Graphics Recognition, Methods and Applications, pp. 35-48, 1996.

[5] H. Cao, R. Prasad, and P. Natarajan, *A stroke regeneration method for cleaning rule-lines in handwritten document images*, MOCR, pages 1-10, 2009.

[6] Z. Shi, S. Setlur and V. Govindaraju, *Removing Rule-lines From Binary Handwritten Arabic Document Images Using Directional Local Profile*, Intl. Conf. Pattern Recognition, pp. 1916–1919, 2010

[7] J. L. Chen and H. J. Lee, *An Efficient Algorithm for Form Structure Extraction Using Strip Projection*, Pattern Recognition, vol. 31, no. 9, pp. 1353–1368, 1998.

[8] J. Said, M. Cheriet and C. Suen. *Dynamical morphological processing: a fast method for base line extraction*, Intl. Conf. on Document Analysis and Recognition, pp. 8-12, 1996.

[9] P. Viola and M. Jones, *Robust Real-Time Face Detection*, Int. J. Comput. Vision, pp. 137-154, 2004

[10] Y. J. Lee, O. L. Mangasarian, *RSVM: Reduced Support Vector Machines*, SIAM International Conference on Data Mining, 2001

[11] C. Cortes, V. N. Vapnik, *Support vector networks,* Machine Learning, **20** pp. 273-297, 1995

[12] J. Limy, D. Rossz, R. Liny, M. Yang, *Incremental Learning for Visual Tracking,* NIPS, pp. 793–800, 2004

[13] J. Kumar and D. Doermann, *Random Subspace-based Data Selection for Large Scale Support Vector Learning*, Pattern Recognition (submitted)

[14] C. Chang and C. Lin, *LIBSVM : a library for Support Vector Machines*, 2001, http://www.csie.ntu.edu.tw/ cjlin/libsvm

[15] DATASET: Rule-line removal in Handwritten Arabic Document Images, Laboratory for Language and Media Processing, http://lampsrv02.umiacs.umd.edu/projdb/