

# Signature-Based Retrieval of Scanned Documents Using Conditional Random Fields

Harish Srinivasan and Sargur Srihari

**Summary.** In searching a large repository of scanned documents, a task of interest is that of retrieving documents from a database using a signature image as a query. This chapter presents a signature retrieval strategy using document indexing and retrieval. Indexing is done using (i) a model based on Conditional Random Fields (CRF) to label extracted segments of scanned documents as Machine-Print, Signature and Noise, (ii) a technique using support vector machine to remove noise and printed text overlapping the signature images and (iii) a global shape-based feature extractor that is computed for each signature image. The documents are first segmented into patches using a region growing algorithm and the CRF based model is used to infer the labels of each of these patches. The robustness of the method is due to the inherent nature of modeling neighboring spatial dependencies in the labels as well as the observed data using CRF. The model parameters are learnt using conjugate gradient descent with line search optimization to maximize pseudo-likelihood estimates and the inference of labels is done by computing the probability of the labels under the model with Gibbs sampling. A further post processing of the labeled patches yields signature regions which are used to index the documents. Retrieval is performed using a matching algorithm to compare the query with the indexed documents. Signature matching is based on a normalized correlation similarity measure using global shape-based binary feature vectors. The end-to-end system is a content-based image retrieval system designed for signatures.

## Introduction

Retrieving relevant documents from a repository of scanned documents has many applications including the legal and forensic domains. In particular documents containing handwriting have a potentially useful role in counter-terrorism operations, e.g., retrieving forms filled out by certain applicants for opening post-office boxes, identifying envelopes of interest in the mail stream, etc. In searching complex documents, a task of relevance is relating the signature in a given document to the closest matches within a database of documents; this is the signature retrieval task which is addressed in this chapter.

Retrieval of handwritten words has been found to be more challenging than image matching due to the lack of low level distinguishing features like color and texture. Handwritten word retrieval has been discussed in Rath et al. (2004), Zhang et al. (2004), Kolz et al. (2000), Plamondon and Lorette (2000). The method of Kolz et al. (2000) extracts profile-based holistic shape features from a line or word image and uses dynamic time warping (DTW) to match words. A word shape based method was shown to perform better than the DTW method, in terms of efficiency and effectiveness (Zhang et al. 2004). Considering historical manuscripts, Rath et al. (2004) describe a method for retrieval based on text queries without recognition using a transcribed set of pages for training.

This chapter presents an effective signature extraction and retrieval technique. It is based on a statistical model for machine learning known as Conditional Random Fields (CRFs) (Lafferty et al. 2001; Kumar and Hebert 2003; Quattoni et al. 2005). CRFs are more general than Hidden Markov Models in that there are no implicit independence assumptions. The CRF model is used in extracting signatures from complex documents by isolating the different contents present in the documents. The motivation to use a CRF based model for this application arises from the spatial inter-dependencies of the different regions in documents. The problem is formulated as follows: Given a document: (i) Segment the document into a number of patches (approximately the size of a word), and (ii) Label each of the segments as one of Machine-Print, Handwriting or Noise. Then the region containing the signatures are identified from the labeled patches and isolated.

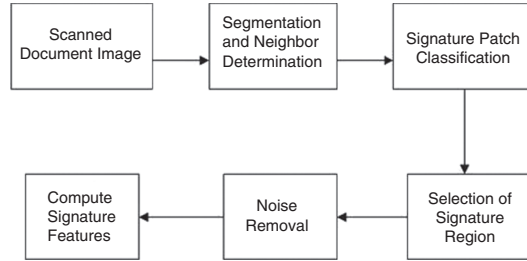
Given a database of signed documents, the retrieval task (Srihari et al. 2006) is to relate a query document to other documents in this database which have been signed by the same author. The documents under consideration are indexed by the features of the signatures extracted from the documents. The retrieval task would be to retrieve all the other documents signed by the same author. This involves extracting the features of the query signature and matching these features to those of the indexed documents. A technique based on query expansion using automatic relevance feedback (Salton and McGill 1983) has also been implemented, where the highest ranked result is used along with the original query to retrieve relevant documents. This retrieval technique can be extended to accept a text query of the authors name provided each author has been previously enrolled with at least 1 signature.

## Indexing

The steps involved in indexing the document images are described here.

### Signature block location

The first step in indexing a scanned document image is to extract the signature block. A signature block is defined as a rectangular image snippet



**Fig. 2.1.** Block diagram of indexing the documents.

containing the entire signature. The signature block is further processed to remove non-signature material, e.g., printed name of the signatory, portions of the accompanying text, spots, etc. The operational steps in signature extraction (Fig. 2.1) are: (i) segmentation into patches and neighbor determination, (ii) classification of patches into signature and non-signature classes, (iii) isolating the signature region (image snippet) from the rest of the image, (iv) removal of noise and printed text from the signature region and (v) extraction of features required for signature matching.

### Segmentation and neighbor determination

A patch is defined to be a region in a document such that, if a rectangular window (size determined dynamically for each document) is drawn with each foreground pixel within the patch at its center, then the window shall not contain any foreground pixel from another patch. The size of the patch was optimized in a way to represent approximately the size of a word. The algorithm for generating these patches is a region growing algorithm and a brief description is given below.

1. Initialize every pixel to be a separate patch.
2. Start with a foreground pixel that is not already marked.
3. With this pixel as the center, draw a rectangular window of size proportional to the height and width of the document being considered.
4. All foreground pixels of connected components with any pixel enclosed within this rectangular window are marked as belonging to the same patch as that of the center pixel.
5. Repeat steps 2 through 4 until all pixels are marked.
6. Patches with pixels lesser than a fixed threshold are ignored as noise and are not attempted to be labeled as one of machine-print, handwriting/signature, noise.

Once all the patches are obtained for a document, the neighboring patches are identified. A total of 6 neighbors are identified for each patch. These neighbors are the closest (top/bottom) and the two closest (left/right) in terms of the convex-hull distance between the patches considered. The reason for including more neighbors from the right and left, is the fact that scanned documents have greater dependency across the width of the document. The definitions of top, bottom, left and right are determined from the center of gravity of the patch being considered. However the convex-hull distance between two patches is measured taking the entirety of both the patches.

### Conditional Random Field model description

A model based on Conditional Random Fields is used to label each of the patches identified using the labels of the neighboring patches. The probabilistic model of the Conditional Random Field used is given below.

$$P(\mathbf{y}|\mathbf{x}, \theta) = \frac{e^{\psi(\mathbf{y}, \mathbf{x}; \theta)}}{\sum_{\mathbf{y}'} e^{\psi(\mathbf{y}', \mathbf{x}; \theta)}} \quad (2.1)$$

where  $\mathbf{y} \in \{\text{Machine-print, Handwriting, Noise}\}$  and  $\mathbf{x}$  : Observed document and  $\theta$  : CRF model parameters. It is assumed that a document is segmented into  $m$  non-overlapping patches. Then

$$\psi(y, x; \theta) = \sum_{j=1}^m \left( A(j, y_j, \mathbf{x}; \theta^s) + \sum_{(j,k) \in E} I(j, k, y_j, y_k, \mathbf{x}; \theta^t) \right) \quad (2.2)$$

The first term in Eq. 2.2 is called the state term and it associates the characteristics of that patch with its corresponding label.  $\theta^s$  are called the state parameters for the CRF model. Analogous to it, the second term, captures the neighbor/contextual dependencies by associating pair wise interaction of the neighboring labels and the observed data.  $\theta^t$  are called the transition parameters of the CRF model.  $E$  is a set of edges that represent the neighbors of a patch.

The association potential can be modeled as

$$A(j, y_j, \mathbf{x}; \theta^s) = \sum_i (h_i \cdot \theta_{ij}^{s_2})$$

where  $h_i$  is typically the state feature value associated with the patch being considered. In order to introduce a non-linear decision boundary we define  $h_i$  to be a transformed state feature vector

$$h_i = \tanh \left( \sum_l (f_l^{s_1}(j, y_j, \mathbf{x}) \cdot \theta_l^{s_1} i) \right)$$

where  $f_l^s$  is the  $l$ th state features extracted for that patch. The state features that are used for this problem are defined later in Table 2.1. The state features,  $f_l$  are transformed by the tanh function to give the feature vector  $\mathbf{h}$ . The state parameters  $\theta^s$  are a union of the two sets of parameters  $\theta^{s_1}$  and  $\theta^{s_2}$ .

The interaction potential  $I(\cdot)$  is generally an inner product between the transition parameters  $\theta^t$  and the transition features  $f_t$ . To introduce non-linearity, we use the idea of kernels, and the interaction potential is defined as follows:

$$I(j, k, y_j, y_k, \mathbf{x}; \theta^t) = \sum_l (\phi_l \cdot \theta_l^t)$$

where  $\phi_l$  is the  $l$ th transition feature after applying a quadratic kernel on the original transition features as defined below.

$$\Phi_l = \langle f^t(j, k, y_j, y_k, \mathbf{x}) \cdot f^t(j, k, y_j, y_k, \mathbf{x}) \rangle$$

**Table 2.1.** Description of the 23 state features used.

State Feature	Description
Height	Maximum height of the patch
Avg component width	The mean width of the connected components within a patch
Density	Density of foreground pixels within the patch
Aspect ratio	Width/Height of the patch
Gabor filter	8 features capturing the different stroke orientations
Variation of height	Variation in height within a patch
Width variation	Variation in width within a patch
Overlap	Sum of overlap in area between the connected components within a patch
Percentage of text above	Relative location of the patch with respect to the entire document
Number of components	Count of the connected components within a patch
Maximum component size	Maximum size of a component within a patch
Points in convex hull	Number of points in the convex hull of the patch
Maximum run length	The maximum horizontal run length within a patch
Avg run length	The average horizontal run length within a patch
Horizontal Transitions	A count of the number of times the pixel value transitions from white to black horizontally
Vertical Transitions	A count of the number of times the pixel value transitions from white to black vertically

### Parameter estimation

There are numerous ways to estimate the parameters of this CRF model (Wallach 2002). In order to avoid the computation of the partition function we learn the parameters by maximizing the pseudo-likelihood of the documents, which is an approximation of the maximum likelihood value. We estimate the Maximum pseudo-likelihood parameters using conjugate gradient descent with line search optimization. The pseudo-likelihood estimate of the parameters  $\theta$  are given by Eq. 2.3:

$$\hat{\theta}_{ML} \approx \arg \max_{\theta} \prod_{i=1}^M P(y_i | y_{\mathcal{N}_i}, \mathbf{x}, \theta) \quad (2.3)$$

where  $P(y_i | y_{\mathcal{N}_i}, \mathbf{x}, \theta)$  (Probability of the label  $y_i$  for a particular patch  $i$  given the labels of its neighbors,  $y_{\mathcal{N}_i}$ ), is given below.

$$P(y_i | y_{\mathcal{N}_i}, \mathbf{x}, \theta) = \frac{e^{\psi(y_i, \mathbf{x}; \theta)}}{\sum_a e^{\psi(y_i=a, \mathbf{x}; \theta)}} \quad (2.4)$$

where  $\psi(y_i, x; \theta)$  is defined as before in Eq. 2.2.

Note that the Eq. 2.3 has an additional  $y_{\mathcal{N}_i}$  in the conditioning set and hence the factorization into products is feasible as the set of neighbors for the patch form the minimal Markov blanket.

From Eqs. 2.3 and 2.4, the log pseudo-likelihood of the data is given by

$$\mathcal{L}(\theta) = \sum_{i=1}^M \left( \psi(y_i = a, x; \theta) - \log \sum_a e^{\psi(y_i=a, x; \theta)} \right)$$

### Features for signature classification

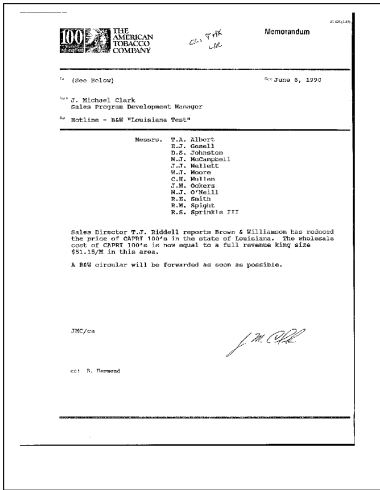
State features try to associate each patch to a label using characteristics of that patch alone. Analogous to these, transition features associate a patch to a label using information from the neighboring patches. Twenty-three state features are extracted for each patch, as described in Table 2.1. Then, the four transition features described in Table 2.2 are computed using the state features and neighbor information. Using these extracted features from each of the 3500 patches in the training set, the parameters of the CRF were estimated as described above. Figure 2.2a shows an example of a document used for feature extraction.

### Classification

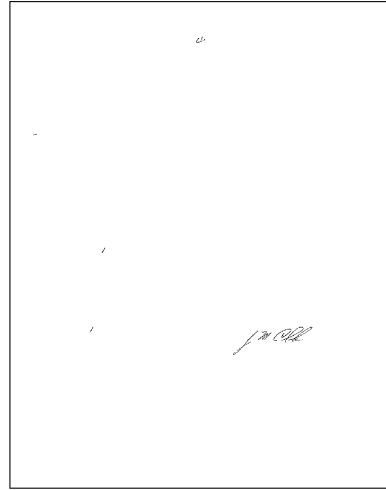
The goal of inference is to assign a label to each of the patches being considered. The algorithm for inference uses the idea of Gibb's sampling (Casella and George 1992).

**Table 2.2.** Description of the 4 transition features used. Transition features are computed for a patch and its neighbor.

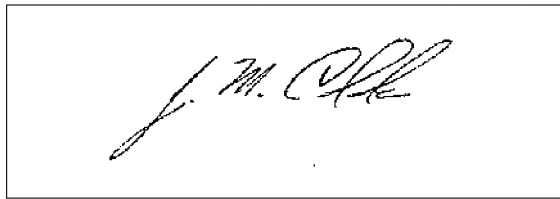
Transition Feature	Description
Relative location	Assigned weights based on the relative location - top/bottom or right/left
Convex hull distance	The convex hull distance between the 2 patches
Ratio of aspect ratio	The ratio of the aspect ratio values of the 2 patches
Ratio of number of components	The ratio of the number of components present in the 2 patches



(a) Original document



(b) Processed document after classification of signature components



(c) Extracted signature

**Fig. 2.2.** Sample signature extraction results (a) Step 1: Feature extraction; (b) Step 2: Classification; (c) Step 3: Post-processing.

1. Randomly assign labels to each of the patches in a document based on an intuitive prior distribution of the labels.
2. Choose a patch at random and compute the probability of assigning each of the labels using the model from Eq. 2.4 to obtain a probability distribution  $p$  for the labels.

3. Use Gibbs sampling to sample from this distribution  $p$  to assign a probable label to the patch.
4. Repeat steps 2 and 3 until the assignments do not change. Store the set of label assignments along with the probability distribution  $p$ .
5. Repeat steps 1–4, for a sufficient number of iterations in order to eliminate the dependency on the initial random label assignments.
6. Consider the set of arrived assignments at step 4 in each of the iterations, and for all the patches pick the labels with the maximum probability as the final set of labels.

Figure 2.2b shows an example of a document image obtained as a result of the classification of the signature labels on the document in Fig. 2.2a.

### Post-processing

In this step, only the patches labeled as possible signatures are considered. Each of these patches is merged with other neighboring possible signature patches, the components on the right and left side being weighed more than those on the top and bottom. A region growing algorithm like the one described above but with a larger window size is used to merge the patches. Other small components which were left out initially are inserted back into the signature blocks being considered. Figure 2.2c shows the result of the post-processing step on the image in Fig. 2.2b.

### Noise removal

Noise removal is carried out to get rid of any noise or printed text overlapping the extracted signature region. We use Support Vector Machines (SVM) (Burges 1998) to classify each connected component as either a part of a signature or a noise component, comprising of printed text, small handwritten text, logos, noise, etc. The SVM is previously trained on the connected components extracted from 10 sample signatures with noise. At the end of the classification step we obtain the signature image with only the signature components remaining. The features used include directional features, height, perimeter and aspect ratio. An example of the results obtained by this noise removal procedure is shown in Fig. 2.3.

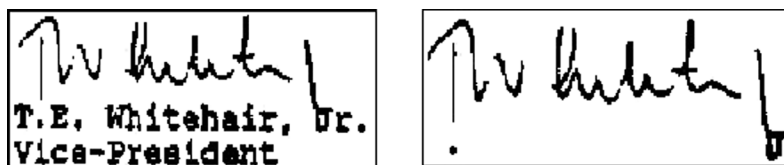


Fig. 2.3. Example of noise removal.

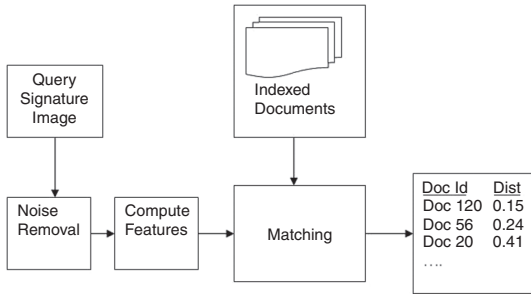




12 rules (Favata and Srikantan 1996). The concavity features capture the major topological and geometrical features including direction of bays, presence of holes, and large vertical and horizontal strokes.

## Retrieval

The document retrieval is performed using a matching algorithm to compare the query with the signature. Figure 2.5 shows the various operational steps in the retrieval process: (i) noise removal from the query signature; (ii) feature extraction from the query signature after noise removal; (iii) matching the query signature features to each of the indexed documents; and (iv) ranking the documents in accordance with the results from the matching algorithm.



**Fig. 2.5.** Block diagram of document retrieval.

## Matching algorithm

Given a query signature image, the relevant documents are retrieved using a matching algorithm. The GSC binary feature vectors are extracted for the query, and the matching algorithm’s task is to compare these features with the indexed features of the signatures present in the database of documents. Figure 2.6 shows a query signature image being matched against a few extracted signatures and the resulting dissimilarity measures obtained using the matching algorithm.

The distance between the queried signature and each of the indexed documents in the database is calculated using a normalized correlation similarity measure (Zhang and Srihari 2003a, b). Given the two binary feature vectors  $X \in \Omega$  and  $Y \in \Omega$ , each similarity score  $S(X, Y)$  uses all or some of the four possible values, i.e.  $S_{00}$ ;  $S_{01}$ ;  $S_{10}$ ;  $S_{11}$ . Here  $S_{ij}$ ,  $(i,j) \in \{0,1\}$ , is the number of occurrences where pattern  $i$  occurs in the first binary vector and pattern  $j$  occurs in the second vector in the same position. The similarity distance  $S(X, Y)$  between two feature vectors  $X$  and  $Y$  is given by Eq. 2.5.

RETRIEVAL	
	<u>Dissimilarity</u>
	0.26
	0.36
	0.39
	0.43
	0.49

**Fig. 2.6.** Subset of retrieval results with the query image on the *left* and the signatures matched against and their corresponding dissimilarity distances on the *right*.

$$S(X, Y) = \frac{1}{2} + \frac{S_{11}S_{00} - S_{10}S_{01}}{2((S_{10} + S_{11})(S_{01} + S_{00})(S_{11} + S_{01})(S_{00} + S_{10}))^{1/2}} \quad (2.5)$$

where

$S_{00}$  = the first binary vector has a 0 and the second vector too has a 0 in the corresponding positions.

$S_{11}$  = the first binary vector has a 1 and the second vector too has a 1 in the corresponding positions.

$S_{01}$  = the first binary vector has a 0 while the second vector has a 1 in the corresponding positions.

$S_{10}$  = the first binary vector has a 1 while the second vector has a 0 in the corresponding positions.

When constructing the similarity distance measure all possible matches  $S_{ij} \in 0,1$  are considered for better classification. Also  $S_{00}$  has been weighted with a beta value of 0.5 to boost classification. The results are ranked in the increasing order of this dissimilarity distance which varies between 0 and 1, a value of 0 indicating an exact match. In the signature retrieval process there is no prior knowledge of the writers signature, the goal is to identify the closest

matching signatures and to identify all the documents containing signatures by the writer of the queried signature. Each of the retrieved signature images is also linked with its corresponding document ID, which allows the user to easily retrieve its location and the document it belongs to.

Before the matching algorithm is applied, the query signature image is processed to remove any overlapping printed or noisy components as mentioned above. Following this, the GSC features for this component are extracted.

### Query expansion using automatic relevance feedback

A query expansion is done using the feedback (retrieval results) of the matching algorithm. The matching score  $S_i$  for a query  $q$ , matched against a document  $D_i$ , given by Eq. 2.6, is computed for each document and sorted in ascending order. The document with the lowest  $S_i$  being the most relevant document retrieved.

$$S_i = S(f(q), f(D_i)) \quad (2.6)$$

where  $f(q)$  is the binary feature vector of the image  $q$ ,  $f(D_i)$  is binary feature vector indexed in  $D_i$ , and  $S(f(q), f(D_i))$  is given by Eq. 2.5.

Let document  $D_i$  correspond to the document with the lowest  $S_i$ . The signature image extracted from the document  $D_i$  is used as a new query  $q_{new}$ , and added to the existing query to formulate an expanded query consisting of the 2 images,  $q$  and  $q_{new}$ .

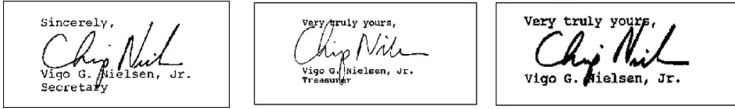
The retrieval is performed using the matching algorithm with this new query  $\{q, q_{new}\}$ . The new score for each document,  $S_i(\{q, q_{new}\}, D_i)$ , is computed by the minimum distance obtained from the 2 queries as given by Eq. 2.7.

$$S_i(\{q, q_{new}\}, D_i) = \min\{S(q, D_i), S(q_{new}, D_i)\} \quad (2.7)$$

This technique improves the accuracy of the retrieved results as the matching algorithm consistently returns relevant documents in the top results.

## Dataset

The dataset used for this experiment was taken from a set of 744 document images signed by 67 different authors. This set of documents consists of a variety of documents, a majority of which have printed text with a signature at the bottom. There are also documents with handwritten text around this printed text, only handwritten documents, documents with images like tables, graphs, etc and multiple signatures per document or no signatures at all. Many of these documents also have logos, other symbolic text and noisy components like words circled or scratched or handwritten text overlapping the printed text or printed text overlapping the signatures. There are also documents with lines and black borders and noise. Some of the writers have several



**Fig. 2.7.** All the automatically extracted samples for writer “10”.

types of signatures like the writer’s full name, initials, only first name, etc. Documents with multiple signatures per document and purely handwritten documents with signatures have also been considered here. For this experiment we randomly picked several different authors and picked 2–5 documents per author making a total of 101 documents containing a total of 114 signatures. Figure 2.7 shows all the signature samples automatically extracted from the documents belonging to one of the writers.

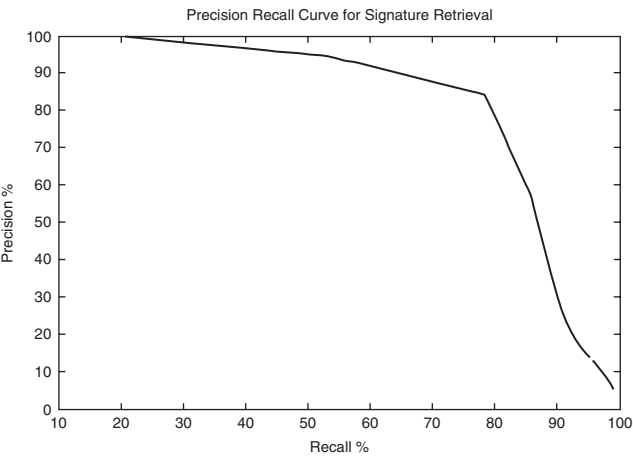
## Experiments and results

In this section, the test setup and the experimental results obtained for the signature retrieval task are described. In the test setup for Signature Retrieval, the images were divided into 2 groups per writer. One group consisting of known document images and the second group consisting of the questioned signatures for testing. The image formats supported are png, jpeg and tiff. The database of documents with known signatures are first processed to index each document. Out of the 101 documents from which the signatures were extracted, in 91.2% (= 104) of the cases the extracted region contained the entire signature image correctly extracted. Following this, the signature image in question is selected and this queried image is preprocessed to remove any overlapping printed text or noise. The set of indexed documents are selected and the signature retrieval process is carried out against this set of known documents. In each case, the precision and recall measures are calculated. The precision and recall measures (Salton and McGill 1983; van Rijsbergen 1979) for a rank “R” where the author of the questioned signature is represented by “A” are defined as follows

$$\text{Recall of label 'a'} = \frac{\text{Amount of correctly classified data of label 'a'}}{\text{Total amount of data of label 'a'}}$$

$$\text{Precision of label 'a'} = \frac{\text{Amount of correctly classified text of label 'a'}}{\text{Total amount of text classified to be of label 'a'}}$$

The testing was done for 1–2 extracted signature images per writer which were randomly selected from the entire set. Each of these signatures was queried against the entire set of 114 indexed signature images in the database. The ranks of the retrieved documents which were signed by the author of the questioned signature were noted in each case and the average precision and recall values were estimated for different ranks.



**Fig. 2.8.** Precision-recall curves for signature retrieval results: Precision of 84.2% at recall of 78.4% after query expansion.

The experiments were conducted using query expansion, where the top results from the retrieval results for the initial query were used along with the initial query to retrieve relevant documents. Figure 2.8 shows the precision recall curves obtained in this experiment. In the top 5 results a recall of 78.4% is obtained, the precision at this point is 84.2%. Table 2.3 shows the results at the end of this phase. There is an increase in the retrieval accuracy on using query expansion, this shows that the system consistently retrieved a relevant document as the top choice. And the usage of this top choice result along with the original query strengthened the retrieval accuracy.

The retrieval accuracy also has been impacted by several factors like: the signature extraction was effective in 91.2% of the cases, so some of the indexed documents contained spurious signature images; the noise removal technique has led to the removal of some components belonging to the signature in a small number of cases; and the poor quality of some of the documents.

**Table 2.3.** Recall measures for signature retrieval from entire database.

No of Results Considered	Recall Measure(%)
Rank 1	18.6
< Rank 3	52.9
< Rank 5	78.4
< Rank 10	87.3
< Rank 15	89.7
< Rank 25	92.6
< Rank 50	97.0

## Conclusions

Here the set of experiments done for the problem of document retrieval using signatures and its results were presented. The tests were conducted on a variety of document and signature samples including those with noise, logos, figures, printed and handwritten text. Although the presence of noise and text overlapping the signatures make retrieval a challenging task, our technique returned a relatively high precision and recall accuracy of 84.2% and 78.4% respectively when considering the top 5 results. This can be attributed to the usage of conditional random fields for the removal of printed and noisy data from the documents leading to an accurate signature extraction in most cases, followed by the usage of an effective matching algorithm using global shape-based features.

## References

- Burges, C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.
- Casella, G. and E. George. 1992. Explaining the Gibbs sampler. *The American Statistician*, 46:167–174.
- Favata, J. T. and G. Srikantan. 1996. A multiple feature resolution approach for handprinted digit and character recognition. *International Journal of Imaging Systems and Technology*, 7:304–311.
- Kolz, A., J. Alspecter, M. Augustijn, R. Carlson, and G. V. Popescu. 2000. A line-oriented approach to word spotting in handwritten documents. *Pattern Analysis and Applications*, 2(3):153–168.
- Kumar, S. and M. Hebert. 2003. Discriminative fields for modeling spatial dependencies in natural images. *Advances in Neural Information Processing Systems (NIPS-2003)*.
- Lafferty, J., A. Macallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequential data. *Eighteenth International Conference on Machine Learning (ICML-2001)*.
- Plamondon, R. and G. Lorette. 2000. On-line and offline handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(1):63–84.
- Quattoni, A., M. Collins, and T. Darrel. 2005. Conditional random fields for object recognition. *Advances in Neural Information Processing Systems 17 (NIPS 2004)*.
- Rath, T., R. Manmatha, and V. Lavrenko. 2004. A search engine for historical manuscript images. *Proceedings of the 27th Annual Int'l SIGIR Conference*.
- Salton, G. and M. J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.
- Srihari, S., S. Shetty, S. Chen, H. Srinivasan, and C. Huang. 2006. Document image retrieval using signatures as queries. *Document Image Analysis for Libraries (DIAL'06)*.
- van Rijsbergen, C. J. 1979. *Information Retrieval*. Butterworths, London.
- Wallach, H. 2002. Efficient training of conditional random fields. *Proceedings of 6th Annual CLUK Research Colloquium*.

- Zhang, B. and S. Srihari. 2003a. Binary vector dissimilarity measures for handwriting identification. *SPIE, Document Recognition and Retrieval X*, pp. 155–166.
- Zhang, B. and S. Srihari. 2003b. Properties of binary vector dissimilarity measures. *Cary, North Carolina*, September.
- Zhang, B., S. N. Srihari, and C. Huang. 2004. Word image retrieval using binary features. *Document Recognition and Retrieval XI, SPIE, San Jose, CA*.





<http://www.springer.com/978-3-642-01140-5>

Computational Methods for Counterterrorism

Argamon, S.; Howard, N. (Eds.)

2009, XVIII, 306 p., Hardcover

ISBN: 978-3-642-01140-5